Computational Models for Image Contour Grouping

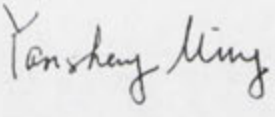
Yansheng Ming

A thesis submitted for the degree of
Doctor's Degree of Philosophy
The Australian National University

March 2015

Final Copy – 24 March 2015

Except where otherwise indicated, this thesis is my own original work.



Yansheng Ming
24 March 2015

To those who inspired me.

Acknowledgments

Pursuing a PhD overseas can be challenging. At the end of this long journey, I must thank those who helped me out in the past four years.

First of all, I am forever grateful to my primary supervisor Dr. Hongdong Li for his guidance and support throughout my PhD candidature. I meet Hongdong on daily basis. He is an excellent supervisor and has the best interests of students at heart. He gives me inspiration and encouragements when I face difficult research problems, and kindly points out right directions when necessary. I am especially impressed by how he fosters basic ideas into high-quality research work. Hongdong not only helps me to produce this thesis, but also cares about my overall wellbeing. In particular, I appreciate his full support for my job hunting.

I am deeply grateful to my co-supervisor Dr. Xuming He. His insights and methods of computer vision research have opened my eye to this fantastic field. In particular, I was greatly inspired by his methodology of establishing new research directions. I also appreciate his advice on good paper writing.

I would like to thank my advisor Dr. Stephen Gould. I learned convex optimization from his well-organized lectures. In addition, I appreciate his timely advice at critical stages of my research.

I must also thank my colleagues who have contributed to my research in various ways. They are Jun Sun, Srimal Jayawardena, Gao Zhu, Mehrtash Harandi, Pan Ji, Yuhang Zhang, Yuchao Dai, Lingqiao Liu. Many of them have kindly helped with revision of my draft papers, offered inspiring discussions, or assisted me with experiments. I feel privileged and proud to be a member of ANU/NICTA computer vision community.

Last but not least, I owe a great deal to my friends: Lin Gu, Shi Wang, Ting Cao, Junming Wei, Yun Hou, Biao He, Jing Guo, Fisher Yu, Yi Li. Their company and friendship added so much fun to my life. In particular, I will cherish the laughters at weekly university house diners. I wish them all the best.

Abstract

Contours are one dimensional curves in images which may correspond to meaningful entities such as object boundaries. Accurate contour detection will simplify many vision tasks such as object detection and image recognition. Due to the large variety of image content and contour topology, contours are often detected as edge fragments at first, followed by a second step known as "contour grouping" to connect them. Due to ambiguities existing in local image patches, contour grouping is considered an essential step for constructing globally coherent contour representation.

This thesis aims to group contours so that they are consistent with human perception. We draw inspirations from Gestalt principles, which describe perceptual grouping ability of human vision system. In particular, our work is most relevant to the principles of closure, similarity, and past experiences. If these principles can be coded into mathematical models, the outputs of these models should be closer to what human see from images.

The first part of our contribution is a new computational model for contour closure. Most of existing contour grouping methods have focused on pixel-wise detection accuracy and ignored the psychological evidences for the importance of topological correctness. This chapter proposes a higher-order CRF model to achieve contour closure in the contour domain. We also propose an efficient inference method which is guaranteed to find integer solutions. Tested on the BSDS benchmark, our method achieves a superior contour grouping performance, comparable precision-recall curves, and more visually pleasant results. Our work makes progresses towards a better computational model of human perceptual grouping.

The second part is an energy minimization framework for salient contour detection problem. Region cues such as color/texture homogeneity, and contour cues such as local contrast, are both useful for this task. In order to capture both kinds of cues in a joint energy function, topological consistency between both region and contour labels must be satisfied. Our technique makes use of the topological concept of winding numbers. By using fast method for winding number computation, we find that a small number of linear constraints are sufficient for label consistency. Our method is instantiated by ratio-based energy functions. Due to the integration of both region and contour cues, our method obtains improved results. User interaction can also be incorporated to further improve the results.

The third part of our contribution is an efficient category-level image contour detector.

The objective is to detect contours which most likely belong to a prescribed category. Our method, which is based on three levels of shape representation and non-parametric Bayesian learning, shows flexibility in learning from either human labeled edge images or unlabelled raw images. In both cases, our experiments obtain better contour detection results than competing methods. In addition, our training process is robust even with a considerable size of training samples. In contrast, state-of-the-art methods require more training samples, and often human interventions are required for new category training.

Last but not least, in Chapter 7 we also show how to leverage contour information for symmetry detection. Our method is simple yet effective for detecting the symmetric axes of bilaterally symmetric objects in unsegmented natural scene images. Compared with methods based on feature points, our model can often produce better results for the images containing limited texture.

Contents

List of Figures

Introduction

1.1 Thesis Statement

Contour grouping is a fundamental problem in both human vision system and computer vision systems. Human can effortlessly group related contours together. Although psychological studies have shown that this process is influenced by Gestalt principles such as the principle of closure, region homogeneity and object familiarity, their underlying computational mechanisms are largely unknown. This thesis suggests three contour grouping computational models which efficiently implement some of these by-and-large principles, and obtains improved results over previous methods. Some applications of contour grouping are also discussed.

1.2 Perceptual grouping and contour grouping

When we look at the world, visual information is firstly coded by numerous receptors on the retina. However, we do not perceive the world as merely colorful dots. Instead, we see lines, surfaces, and objects, and scenes. The process for this vision function is called perceptual grouping. Scientists investigate this process since the earliest days of vision research. In fact, perceptual grouping phenomena have fueled the Gestalt school of psychophysics. The development of Gestalt theory has impacted the general theory of human perception [Koffka, 1935].

Perceptual grouping is of great interest to computer vision research as well. Many computer vision problems such as image segmentation, contour grouping, and symmetry detection can be considered as subproblems of perceptual grouping. Satisfactory solutions to these problems will shed light on vision mysteries and lead to many applications. In addition, perceptual grouping is also tightly connected to many other vision problems such as object recognition and detection. For example, segmented images regions have been used as basic units for scene understanding.

This thesis focuses on contour grouping problem. Contours are physically-meaningful one

dimensional curves in images. They may correspond to discontinuities in depth, discontinuities in surface orientation, illumination change, or surface markings [Marr, 1982]. Due to their physical meanings, contours can provide a lot of information for a vision system. On a high level, a vision system needs to answer two fundamental questions: where and what. Detection of contours could help answer both of them. For example, the boundary of an object provides precise information about the object's shape and location. Surface markings may also have semantic information, e.g. stripes of a zebra are characteristic of this species. Therefore, it is not surprising that contour detection/grouping underlies so many interesting applications. For example, contour cues are used for computing pairwise pixel affinity in a popular segmentation algorithm [Shi and Malik, 2000b]. The Canny edge detector was employed for the sketch-based image retrieval problem [Eitz et al., 2011]. An image detection method took advantage of contour features to achieve state-of-the-art results [Shotton et al., 2008]. Contours can also be used for tracking [Liu et al., 2009].

Contour grouping is one of the earliest computer vision problems, dated back to early endeavors [Marr, 1982]. Over past a few decades, the performance of detectors has improved steadily. However, there is still a huge gap between human performance and computers' performance, according to benchmarks on the BSDS datasets [Arbelaez et al., 2011]. In addition, computer vision algorithms often overlook other important aspects of contour grouping, e.g. the topological correctness of contours. Besides, human vision system can easily differentiate object boundaries from less important ones. However, this task is difficult for a computer based contour detector.

Our insight is that if machine could mimic known properties of human vision system, the performance disparity between human and machine may be reduced. We pay special attention to some psychological findings such as Gestalt principles which reveal a great deal of why human see certain visual patterns. Next section reviews several well-known Gestalt principles.

1.3 Gestalt principles

The Gestalt school of psychology proposed influential theories for understanding human perception, in particular, vision. In contrast to structuralism, Gestaltism maintains that "the whole is other than the sum of parts." Therefore, the most important problem in this school is to explain the formation of global perception. In particular, several classic Gestalt principles are proposed for describing how the human visual system could build global structures from elements. Figure 1.1 illustrates some of these principles.

Proximity: The Gestalt principle of proximity states that patterns which are close to each

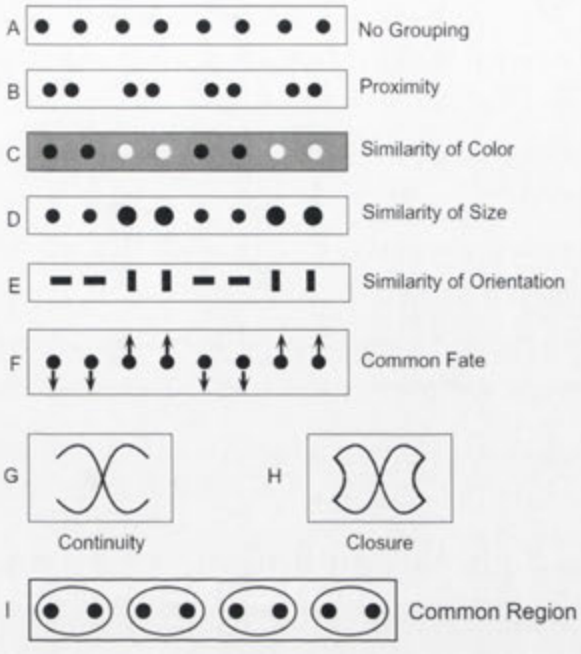


Figure 1.1: An illustration of some Gestalt principles, reproduced from [Palmer et al., 2003].

other are preferred to be grouped together given other conditions equal, as shown in Figure 1.1(B)

Similarity: The Gestalt principle of similarity states that patterns which are visually similar are grouped together given other conditions equal. Similarity can include similarity in color, size and orientation shown in Figure 1.1(C)(D)(E) respectively.

Closure: The Gestalt principle of closure states that human prefer to see complete figures instead of incomplete ones if other conditions are equal. This principle is demonstrated in Figure 1.1(H).

Good continuity: The Gestalt principle of good continuity means that human prefer to see smooth contours rather than non-smooth ones, if other conditions are equal. As demonstrated in Figure 1.1(G), human usually see two continuous curves crossing each other.

Past experience: The Gestalt principle of past experience states that patterns which have been seen many times before are preferred over unfamiliar ones, given other conditions equal.

Common region: The Gestalt principle of common region states that patterns enclosed by a common region are more likely to be considered as a whole, given other conditions equal, as shown in Figure 1.1(I).

Common fate: The Gestalt principle of common fate states that patterns which are moving together are more likely to be considered as a whole, given other conditions equal. As shown

in Figure 1.1(F), the elements which are moving together are preferred to group together.

These principles are interesting because they relate mysterious visual perception to physical properties of stimuli such as color, motion and geometry. Therefore, it is tempting to make computer algorithms follow these principles. However, implementing these principles in computers is not straightforward. For example, since human vision system is constructed based on neurons, do our methods have to simulate a neural network? If not, in what sense can different findings be combined in one theoretic framework? To address these problems, we employ the celebrated framework of image understanding established by Marr. In the next section, we briefly review key components of Marr's research and introduce some concepts our approach will use.

1.4 Marr's paradigm for multi-disciplinary research

At the beginning of vision research, the endeavour to understand visual perception was carried out in different disciplines. Psychologists such as Wertheimer proposed that perception was mainly about extracting global properties of stimuli rather than local ones. A major aspect of this school was the discovery of Gestalt principles for perceptual grouping. These principles describe human preference for certain visual patterns [Palmer et al., 2003]. In contrast, physiologists tried to understand vision on a cellular level, by studying neurons in visual cortices. For example, Hubel and Wiesel discovered simple cells and complex cells which had selective responses to certain stimuli [Hubel and Wiesel, 1959]. Vision problems also attracted scientists in AI areas. Their main interests are to build vision machines that can replace human. Despite early optimism, most of vision tasks which appeared to be easy turned out to be difficult to solve. As a result, many researchers retreated to study special mini-worlds or resorted to empirical approaches.

To integrate results from different disciplines, Marr proposed that it was essential to understand vision from the perspective of information processing. In other words, one needs to make clear what is computed, how it is computed, and the physical foundation of the computation. Finally, one can exam the physical implementation of a vision system. His framework made it possible to compare quite different vision systems, e.g. the visual pathway and a computer algorithm, on a computational level, and to use knowledge in separate areas.

Process and **Representation** are two central concepts in Marr's theory. A "process" refers to the transformation or mapping of information. The key part of a process is its "representation". It is a formal system for making explicit certain entities or types of information, together with a specification of how the system does this. The result of applying a representation is

called **description**. Marr argued that a process should be understood at three different levels.

The top level is **computational theory**, this level makes clear the input and output of the process, as well as the constraints/assumptions used in the information mapping.

The middle level is **representation and algorithm**. This level is about finding suitable representations for both input and output of the process. Also, the algorithm which carries out the mapping between two representations are determined on this level.

The bottom level is **hardware and implementation**. This level determines the physical implementation of its representation and algorithm. In the computer vision area, algorithms are usually implemented on Silicon-based computers. In contrast, the computation of the human vision system is carried out by neurons.

Marr's theory laid down the foundation for our research. It suggests that human and computer vision systems are both instances of visual information processes. They can share the same computational theory, and at the same time have different implementation. The key issue is to find suitable representations for these by-and-large principles. In the next section, we will discuss how to represent several Gestalt principles of interest.

1.5 Modeling Gestalt principles

The contour grouping methods proposed in this thesis can be best understood from the perspective of Marr's paradigm. On a computational level, our methods mainly draw on the Gestalt principles. They describe essentially how an input natural image should be grouped into objects. In other words, assumptions made by the human vision system can be derived from these Gestalt principles. The representations and algorithms of our methods are based on state-of-the-art machine learning techniques.

Among all Gestalt principles, we are interested in the principles of closure, similarity, and past experience [Palmer, 1999], and consider them as constraints. For example, the principle of closure says that the contours which form a closed figure are preferred to be grouped together, given other conditions equal. This principle reflects the observation that object boundaries, especially those of foreground objects, appear as closed contours in images. On the other hand, the open contours usually correspond to surface markings of less importance.

However, it is important to note that these principles can not be directly translated into computer codes. First of all, these principles are often based on concepts which are not defined mathematically. Take the Gestalt principle of similarity for example, it is not clear how we determine whether two patterns are similar or not. Second, every principle assumes that all the conditions of two possible perceptions are equal except the one under consideration. However,

it is rarely the case for natural scene images. Third, there are exceptions to every principle. For example, an object may be partly occluded, and not satisfying the principle of closure. It is also possible that part of its boundary is too faint to be completed.

To deal with these challenges, our overall strategy is to carefully design representations and algorithms based on cutting-edge machine learning methods such Conditional Random Field (CRF) [Lafferty et al., 2001], graph partition frameworks, and Gaussian Process Latent Variable model (GPLVM) [Lawrence, 2005]. CRF and the closely related Markov Random Field (MRF) have been wildly used for modeling image statistics. In particular, CRFs have been successfully applied to stereo vision, image segmentation, semantic labeling and many other problems. Recent development is to employ high-order potential functions to model complex label relationship [Gould, 2012]. CRF will be used for modeling contour closure in Chapter 4. Graph partition methods, such as normalized cuts, ratio cuts, and ratio contour are wildly used in perceptual grouping problems. These methods are related to our method in Chapter 5. GPLVM is a Bayesian nonparametric method for dimension reduction. It is known for the ability to model non-linear relationship between observation and low-dimensional latent variables. Our method for category specific contour detection rely on GPLVM for non-linear dimension reduction. In Chapter 3, we will introduce these techniques in detail.

Although our methods relies on Marr's classic theory of image understanding, we would like to clarify important difference in shape representations. Marr's framework starts with input images. The second level of his framework is a **primal sketch** which encodes all the important information in 2D images, including zero-crossing, blobs terminations, edge segments and boundaries, etc. Then the framework progresses to a **2.5D** sketch in which the depth and surface orientations are computed. However, these information is still organized in a viewer-centric coordinate frame. Finally, we arrive at a **3D model representation** which describes 3D shapes in object-centered frame using volumetric primitives. In this framework, recognition is mainly based on 3D information. Our thesis deviates from his framework in two ways. First, semantic information can be derived from 2D information instead of 3D information in our methods. Second, rather than stage-wise and symbolic computation, the representations in our methods are usually probabilistically and recurrent. In other words, our methods do not make hard decisions at early stage of computation.

To summarise, our approach, a combination of classical Gestalt principles and machine learning techniques, is applied to three contour grouping scenarios. We have found that our approach leads to computational models with interesting properties. Our models are capable of extracting highly completed contours from images, improving the robustness of contour detection by incorporating region cues and using prior information to differentiate object contours

from those of background. See Chapter 1.6 for details.

1.6 Thesis Outline

The remaining chapters of this thesis are summarized as follows:

Chapter 2 surveys various methods used for the contour grouping problem. This chapter also discusses closely related problems such as image segmentation, object detection and symmetry detection. The connections between contour grouping and these problems are outlined.

Chapter 3 presents a contour completion model which respects the contour closure effect described in the Gestalt school of psychology. We will first describe our representation, which consists of three kinds of edge proposals extracted from images. Then, we will focus on designing an energy function such that the Gestalt law of closure could be approximated in an efficient way. The details of learning and inference processes will be discussed. This model is extended to detect junctions in natural scene images at the end of this chapter.

Chapter 4 exploits the interaction between contour grouping and image segmentation. Considering this problem in an energy minimization framework, this chapter presents energy functions which take both contour cues and region cues into account. The key assumption of these energy functions is the consistency of edge labels and region labels. Our key contribution, the winding number constraints, is able to ensure the label consistency with low computational costs. We show that our framework improves the accuracy of salient contour grouping on two datasets. Last, we will discuss how to extend this method to interactive segmentation.

Chapter 5 focuses on the detection of category-specific contours, utilizing top-down information. A three-layer representation for contour maps is firstly discussed. Based on this representation, an energy function which couples variables on each level is presented. We will discuss how our model learns contour distribution from either groundtruth edge images and images from the same category. We will also discuss our efficient inference algorithm for this problem.

Chapter 6 shows how contour grouping can help detect symmetric patterns in natural images. This chapter first discusses the extraction of symmetric edge pairs from images and connecting them into a graph of contextual interaction. Then, this chapter discusses our formulation of the symmetry detection problem as finding maximal weight star-subgraphs.

Chapter 7 gives the conclusion and discusses possible future research directions.

Background and related work

This chapter gives the background of this thesis. Section 2.1 presents the scope of each chapters. It also discusses related work and outline the new features of our methods. The next three sections are introductions to machine learning methods our methods used in the following chapters, and readers can skip these sections if they are already familiar with these techniques. Section 5.3 introduces basic concepts and techniques of conditional random field. We constructed CRF models for modeling closure and category-specific priors in Chapter 3 and Chapter 6, respectively. Section 2.3 provides a brief introduction to graph partition frameworks. This section provides the technical background for our method in Chapter 4. Section 2.4 introduces the Gaussian process latent variable model. This section is necessary for understanding our method for detecting category specific contours in Chapter 6.

2.1 Scope of this thesis

In Marr' theory, contours are obtained at the stage of deriving a full primal sketch [Marr, 1982; Guo et al., 2007]. However, more recent studies show that the perceptual grouping processes are not likely to be completed in one shot [Palmer et al., 2003]. Various information can contribute to solving the contour grouping problem at different stages. This thesis presents several contour grouping models in several scenarios. In this section, we discuss closely related work.

Local texture is one of the most popular cues for extracting contour fragments. Classical methods often apply filters such as the Sobel operator to an image, and seek the local maxima of filter responses as edge points [Canny, 1986; Marr and Hildreth, 1980; Jain, 1989]. In addition, the phase information of filter responses was also used in [Kovesi, 1999]. Learning-based methods have been developed (e.g. [Martin et al., 2004; Piotr Dollar, 2006; Kokkinos, 2010b; Ren and Bo, 2012]). Compared with traditional methods like the Canny detector [Canny, 1986], learning based methods are able to exploit various image cues to achieve higher robust-

ness. Latest experiments show that their performance is quite close to human in a local edge discrimination task [Zitnick and Parikh, 2012]. Since contours in natural images emerge at multiple scales [Lindeberg, 1994], pooling multiple-scale information can also improve contour detection results [Ren, 2008; Arbelaez et al., 2011]. These local methods are foundations of contour grouping methods. Our methods mainly use [Martin et al., 2004] due to its accuracy and popularity.

Psychological phenomena such as illusionary contours [Koffka, 1935] demonstrate that the perception of contours is not purely based on local information. Many methods have been proposed to improve detection accuracy by incorporating large scale, or global cues such as Gestalt principles. Among them, the Gestalt principles of proximity and good continuity are most popular, and they favor smooth transitions with short gaps. Their computational soundness has been proved by an empirical study on the BSDS dataset [Ren et al., 2006]. These principles can be represented by the elastica functionals [Sha'asua and Ullman, 1988; Leung and Malik, 1998; Horn, 1983], or a CRF model [Ren et al., 2008]. Contour saliency methods, especially those considered the facilitation effect [suen Lee and Medioni, 1999; Guy and Medioni, 1993; Tong and Tang, 2005; Li, 1998; Parent and Zucker, 1989] also makes extensive use of these cues. In Chapter 3 and 4, we construct energy terms to reflect these principles.

2.1.1 Contour closure modeling

Chapter 3 is centered on the contour closure property. As a global property, contour closure is not easy to implement, especially in a contour domain. Contour closure can be enforced by restricting a solution to be a single connected contour [Elder and Zucker, 1996; Mahamud et al., 2003; Wang et al., 2005; Williams and Thornber, 1999; Schoenemann et al., 2011; Kass et al., 1987]. However, without modeling multiple contours in the image, these methods are not effective for images with occlusion.

Therefore, it is hypothesized that closure can only be achieved by segmentation, as closed contours can be obtained from boundaries of segmentation [Levinshtein et al., 2010a; Arbelaez et al., 2011]. However, a contour-based method is still desirable due to its connections to biological findings [Palmer, 1999], and the ease for junction representations. In addition, the sparsity of edges (compared with image pixels) can often result in lower computational cost.

To enforce closure in a contour domain, our method encodes topological conditions in a high-order conditional random field. Traditional CRF methods like the Ising model [Niss, 2009] are usually built upon pairwise potentials. Due to the importance of high-order statistics, more and more CRF models include high-order potential functions, e.g. those for stereo vision [Woodford et al., 2008], scene labeling [Gould, 2012] [Ladicky et al., 2012], image de-

noising [Roth and Black, 2005], and segmentation [Lempitsky et al., 2009]. In particular, the topological prior of connectivity has been used for the interactive segmentation problem [Vicente et al., 2008]. Unlike their model, our method does not require all the edgelets to form a connected graph. The inference of these methods is usually done by customized algorithms. However, many general methods have been proposed to reduce high-order cliques to low order ones [Gallagher et al., 2011] [Ishikawa, 2009].

Our method shows that highly-completed contours can be obtained without segmentation. In comparison, most contour completion methods can not guarantee contours to be connected [Ren et al., 2008; Kokkinos, 2010b]. In paper [Andres et al., 2011], contour closure is achieved by an exponential number of linear constraints. Their method relies on segmentation to extract boundary hypotheses. Their inference problem comes down to a large scale integer program. In comparison, our method only involves a constraint set of linear complexity with respect to the number of edges, and admits an efficient inference algorithm.

2.1.2 Using segmentation cues for contour grouping

Chapter 4 presents a model which connects image segmentation methods to contour grouping methods, based on the topological concept of winding numbers [Needham, 1999; Gallier and Xu, 2013]. Unfortunately, a winding number is often confused with another topological concept, the rotation index defined as the total rotation angle of tangent if one travels along a curve [Whitney, 1937]. A rotation index is often called a rotation number, even a winding number in the literature [McIntyre and Cairns, 1993]. In computer vision field, rotation indices have been used for ensuring contour topology in [Elder and Zucker, 1996]. To the best of our knowledge, winding numbers have not been applied in perceptual grouping contexts.

Image segmentation is often considered as the dual problem of contour grouping. Its objective is to partition image pixels into coherent and meaningful regions. For example, figure-ground segmentation methods separate whole objects from the background. Supervised segmentation methods can segment an image into regions and recognize the category of each region [He et al., 2004]. Not using supervised information, our method belongs to the category of unsupervised segmentation methods including active contours [Kass et al., 1987], level set [Sethian, 1999] and watershed [Vincent and Soille, 1991]. We pay special attention to methods belonging to a graph partition framework. By representing image pixels as coupled graph nodes, the image segmentation problem is transformed into a graph partition problem. Notable methods under this framework include max-flow [Boykov et al., 2001], normalized cuts [Shi and Malik, 2000a] and ratio cut [Wang and Siskind, 2003]. Moreover, spectral K-means [Bishop, 2006], Felzenszwalb et al.'s graph-based method [Felzenszwalb and Hutten-

locher, 2004], SWA [Alpert et al., 2012] can also be considered as instances of this framework. In addition, the idea of graph partition also underlies interactive segmentation methods such as [Rother et al., 2004].

Some existing work has attempted to incorporate both region cues and contour cues. Intervening Contour [Leung and Malik, 1998] is one of the early efforts to use local contour saliency for region segmentation. This method is built into a state of the art segmentation algorithm [Arbelaez et al., 2011]. Yu et al. [Yu et al., 2001] incorporate edge information in their Markov random field. Tabb and Ahuja integrate both cues for low-level structure detection [Tabb and Ahuja, 1997]. GPAC [Sumengen and Manjunath] is one of the top variational methods for cue integration in a continuous domain. However their gradient descend inference method is susceptible to local minima. Our method, however, obtains global optimal solutions by solving linear programs. Stahl and Wang [Stahl and Wang, 2007] modify the ratio contour method [Wang et al., 2005] in order to extract more regular shapes. The same ratio objective function is implemented by superpixels in [Levinshtein et al., 2010b], and improved results are obtained. In [Nicolls and Torr, 2010], edge labels are determined by a linear transform of region labels, based on the discrete topology of planar graphs. However, each edge in their method is restricted to a predetermined orientation. For interactive segmentation, [Schoenemann et al., 2012] uses local constraints to achieve boundary-region consistency. In contrast, the winding number constraints in our method are global constraints. The winding number concept not only leads to a smaller number of constraints but also provides a clearer understanding about region contour interaction. Winding numbers have recently been used for extracting volume representations from 3D meshes [Jacobson et al., 2013].

2.1.3 Top-down guided contour grouping

Recent studies show that the superiority of the human perceptual grouping system over computer vision methods is largely in the use of high-level category-specific information [Zitnick and Parikh, 2012]. Chapter 5 presents a method which utilizes high-level top-down information to extract the boundaries of recognizable objects. There are several methods for similar purposes. The inverse object detector [Hariharan et al., 2011] estimates object contours from the outputs of object detectors. A category-specific contour model is learned from training images containing similar objects [Wu et al., 2010b]. Their model was extended to extract contours in 3D [Wu et al., 2010a]. Top-down information can also be incorporated in variational methods [Bresson et al., 2006; Cremers et al., 2003; Prisacariu and Reid, 2011a], in which the objective is to find target shapes.

Most methods using high-level information such as [Ren et al., 2005; Zheng et al., 2010]

learn from clean contour images with the exception of ABM [Wu et al., 2010c] which has demonstrated the ability to learn from weakly supervised data, i.e. the raw images. Our method can also learn from such raw images, and captures greater intra-class variations.

A key issue is modeling category-specific shape variations. A traditional structure-based approach represents a shape as a set of densely sampled points on its contour [Kendall, 1989]. For a new image, this approach needs to match a set of model points to image edge points [Ferrari et al., 2010]. However, it is not convenient to deal with complex structural changes, i.e. appearance or disappearance of object parts.

Our method follows a template/feature-based approach. This approach extracts features from roughly aligned object contour images or shape masks, and models the distribution of features through various machine learning techniques. For example, KPCA [Schölkopf et al., 1998] has been applied to level-set type variational shape representations [Cremers, 2006] [Leventon et al., 2000]. [Prisacariu and Reid, 2011b] used a non-linear dimension reduction technique called GPLVM [Lawrence, 2005] to model the non-linear variations of shape masks. They obtain better results than PCA-based methods. Our method also uses this method for dimension reduction. Different from [Prisacariu and Reid, 2011b], our method is able to handle contours of arbitrary topologies. GPLVM has also been applied to estimating human poses [Ek et al., 2008]. The Shape Bolzmann Machine [Eslami et al., 2012] obtains good results on shape masks by learning a hierarchical representation. The inverse detector method [Hariharan et al., 2011] was proposed to predict edge labels from activations of the poselet detectors [Bourdev and Malik, 2009] based on discriminative learning. Compared with [Hariharan et al., 2011], our method spares the effort of training a complicated pose-let detector whenever a new category is introduced.

2.1.4 Symmetry detection

Chapter 6 applies contour cues to solve the symmetry detection problem. Our world is teemed with symmetric natural patterns and man-made objects. Therefore, it not surprising that symmetry perception is a prominent feature of human vision [Koffka, 1935]. In the computer vision field, symmetry detection has led to applications such as image matching [Hauagge and Snavely, 2012] and image segmentation [Sun and Bhanu, 2012]. Mathematically, symmetry can be defined as an isometry which keeps the input pattern unchanged. Furthermore, all possible symmetries can be categorized into a number of groups [Liu et al., 2010a].

In this thesis, we concentrate on the detection of the most basic kind of symmetry, i.e. reflection symmetry, due to its abundance in natural scene images. We also limit a symmetry axis to be a straight line rather than a curve segment as in [Tsogkas and Kokkinos, 2012;

Levinshtein et al., 2009a].

Our method follows the idea that the reflection symmetry can be detected by grouping matched symmetric features. For example, [Loy and Eklundh, 2006] employed SIFT features to describe symmetric texture patterns. However, there are many objects with little texture but strong contours. We observe that their method does not handle this case well. Similar to [Liu and Liu, 2010; Ishikawa et al., 2005], our method uses graphs to represent the interaction of symmetric elements. [Ishikawa et al., 2005] detects the medial axis of a symmetric shape by finding the most salient tree subgraph in an affinity graph. To better enforce the geometric consistency between elements, our method seeks the optimal star-subgraph instead.

A few symmetry detection models are also contour-based [Prasad and Yegnanarayana, 2004; Stahl and Wang, 2008; YlaJaaski and Ade, 1996]. In particular, our model is close to [YlaJaaski and Ade, 1996] in which pairs of edgelets are also grouped into clusters based on pairwise geometric relationship. However, their method committed to a lot of hard decisions, in order to restrict the number of pairwise connections, to the effect that large symmetric patterns cannot be detected. More recently, symmetric edge trapezoids are grouped into symmetric closed contours [Stahl and Wang, 2008]. Different from their model, our method does not require contours to be closed.

2.2 Conditional random fields

Conditional random field is a machine learning technique which has been widely applied to computer vision problems. In Chapter 3 of this thesis, it is used for modeling the distribution of the connected contours. This section introduces readers to concepts and techniques most relevant to understanding of our work. More comprehensive materials can be found at [Kollar and Friedman, 2009].

A conditional random field (CRF) [Lafferty et al., 2001] is a probabilistic distribution $P(Y|X)$ of random variables $Y = \{y_1 ... y_N\}$, conditioned on the input X. The specialty is that the distribution $P(Y|X)$ respects a set of conditional independence properties specified by an undirected graph $G(V, E)$, where V denotes all nodes which are in one to one correspondence to random variables in Y, and E denotes the edges of the graph. The conditional independence property states that, the probability of a subset of nodes U, conditioned on the input and the rest of the nodes, equals the probability conditioned on the input and the neighbors of U: $P(Y_U|X, Y_{V/U}) = P(Y_U|X, Y_{ne(U)})$, where V/U denotes the set of nodes not in V. The set $ne(U)$ denotes nodes in V/U which are directed connected to any nodes in U by edges in E.

A closely related concept is Markov random fields (MRF) originated as the Ising model [Ising, 1925] (see a textbook [Bishop, 2006] for a modern introduction). The main difference is that a MRF models the joint distribution of X and Y rather than the conditional probability. Usually, the joint probability is more complicated to model. For a semantic segmentation problem, input X is an image and Y is the semantic labels. To build a MRF model, one has to estimate the probabilities of images, which can be quite difficult. However, the conditional distribution of semantic labels is much easier to estimate.

A central concept for both CRF and MRF is a clique. It is defined as a subset of nodes in which any pair of nodes are directly connected in G. A maximal clique is a clique which cannot include any more nodes. Please note that maximal cliques are determined by the graph G. However, the *Hammersley-Clifford* theorem states that $P(Y|X)$ can be factorized according to the cliques:

$$P(Y|X) = \frac{1}{Z}\prod_{C}\Psi_C\left(Y_C|X\right) \tag{2.1}$$

Where Z is a normalization constant, C denotes a maximal clique, and Y_C denotes the variables in this clique. The maximal size of maximal cliques is called the order of a CRF. A high-order CRF has more expressive power than low-order ones. However, it poses challenges to both learning and inference. Up to date, this problem is still an active research area.

The negative logarithm of $P(Y|X)$ is called the *energy function* of this distribution. Maximizing the probability is equivalent to minimizing the energy. In this thesis, our CRFs usually have high orders. However, we make use of special properties of our problems to derive simplified inference and learning procedures. In the following, we briefly introduce popular learning and inference techniques of CRFs.

2.2.1 Inference methods

The goal of CRF inference can be either estimating marginal distributions of variables or a maximum-a-posteriori solution (MAP). Here we focus on finding MAP solutions of a CRF. There are myriads of methods for this purpose, and this chapter only reviews the most relevant ones. The one used by our method in Chapter 3 is based on linear relaxation. This section also discusses graph cuts. To a special kind of energy functions, Graph cuts can obtain the global optimal solutions in polynomial time. Chapter 3 will discuss whether our method can be solved by Graph cuts. Then we introduces the message passing methods which are also relevant to Chapter 3.

2.2.1.1 Graph cuts

A special kind of CRF is quadratic pseudo boolean functions of the following form:

$$E(Y) = \sum_i a_i y_i + \sum_{ij} a_{ij} y_i y_j \tag{2.2}$$

where a_i and a_{ij} are coefficients, and $y_i \in \{0, 1\}$. When all a_{ij} are negative, the energy function is *submodular*. For submodular functions, the problem can be equivalently transformed into a minimal cut problem. This problem in turn can be transformed into a max-flow problem which can be solved in polynomial time [Boykov et al., 2001]. Graphcut can be extended for multiple label CRFs. For non-submodular functions, one can use an extension called QPBO to compute a subset of the optimal labels [Hammer et al., 1984]. There is an extension called QPBO-P which can obtain more labels by probing [Rother et al., 2007]. These methods only solve quadratic functions. However, higher order energy can be provably reduced to low-order ones [Hammer et al., 1984].

2.2.1.2 Linear relaxation

The energy function (2.2) can also be minimized by linear relaxation. First, we need to relax the integer constraints on labels into linear ones, i.e., $0 \leq y_i \leq 1$. The quadratic terms such as $y_i y_j$ can be replaced by a variable t_{ij}, plus some constraints. Together, the linear program is as follows:

$$\min_{Y,T} \sum_i a_i y_i + \sum_{ij} a_{ij} t_{ij} \tag{2.3}$$

$$s.t. \quad 0 \leq y_i \leq 1 \tag{2.4}$$

$$t_{ij} \leq y_i \tag{2.5}$$

$$t_{ij} \leq y_j \tag{2.6}$$

$$t_{ij} \geq y_i + y_j - 1 \tag{2.7}$$

$$t_{ij} \geq 0 \tag{2.8}$$

When Y, T are boolean vectors, one can immediately see that Eq (2.3) is equivalent to Eq (2.2). After relaxation, Eq (2.3) only estimates a lower bound of the true energy in general. However, when the energy function is submodular, the linear relaxation is also tight [Hammer et al., 1984].

2.2.1.3 Message passing

A classical message passing method is belief propagation (BP) [Pearl, 1982]. The BP method for MAP inference is also called the max-product or max-sum algorithm. The message-passing procedure is best presented with a *factor graph*. A factor graph is a bipartite graph. A node in a factor graph may represent a variable or a clique potential Ψ_C in Eq (2.1). There are edges connecting each potential with the associated variables. Then the message passing procedure includes passing messages from variables to factors and from factors to variables.

$$\mu_{\Psi_C \to y}(y) = \max_{y_1,\ldots,y_M} \left(\ln \Psi_C(y, y_1, \ldots, y_M) + \sum_{m \in \mathrm{ne}(\Psi_C) \setminus y} \mu_{y_m \to \Psi}(y_m) \right) \tag{2.9}$$

$$\mu_{y \to \Psi_C} = \sum_{l \in \mathrm{ne}(y) \setminus \Psi_C} \mu_{\Psi_l \to y} \tag{2.10}$$

BP is based on the distributive law. For graphs of special structures such as tree graphs, BP obtains global optimal solutions in polynomial time. For general graphs, it often obtain good solutions. An extension is to decompose a general graph into tree graphs and solve them independently by BP [Kolmogorov, 2006; Wainwright et al., 2005].

2.2.2 Learning methods for CRF

In this section, we introduce two general learning methods for conditional random fields.

2.2.2.1 Maximal likelihood learning

If we consider a probability distribution $P(Y;\Theta)$ with parameters Θ (the input X is omitted here for clarity). Some iid training samples $\{Y_i | i = 1...N\}$ are provided. The maximal likelihood learning seeks the parameters which maximize the likelihood of the training samples.

$$\Theta^* = argmax \prod_i P(Y_i;\Theta) \tag{2.11}$$

This optimization problem is convex if the distribution is log-linear with respect to the parameters. However, the gradient of Eq (2.11) is quite expensive to compute. Methods such as contrastive divergence [Hinton, 2002] are proposed to approximately estimate gradients.

2.2.2.2 Large margin CRF learning

If we consider a CRF as a structured classifier, it natural to extend the well established principle of large margin learning to CRF learning problem [Taskar et al., 2005]. The objective of this approach is to separate groundtruth labels far away from all other possible labels. The benefit is two fold. First, this approach concentrates on learning a robust decision boundary which is easier than learning the whole distribution in maximal likelihood learning. Second, the optimization problem of large margin learning avoids the expensive computation of the partition function. Assume that the energy function is linear with respect to the parameter $E(Y;\Theta) = \Theta^T \mathbf{f}(Y)$, where $\mathbf{f}(Y) = [f_1(Y), ..., f_{N_f}(Y)]$ is a vector of N_f potential functions. The large margin learning is as follows:

$$\min_{\Theta} \frac{1}{2}||\Theta||_2^2 \tag{2.12}$$

$$s.t. \quad \Theta^T \mathbf{f}(Y_i) \geq \max_{Y \in \mathcal{Y}} \left(\Theta^T Y + l(Y_i, Y) \right), \quad \forall i \tag{2.13}$$

where Y_is are groundtruth labels, and l denotes a loss function. The set $\mathcal{Y}$ denotes all possible labels of Y.

2.3 Graph partition methods

Graph partition is key concept underlying many image segmentation algorithms. In chapter 4, our method incorporates image segmentation cues which are represented as a fully connected graph. Therefore, in this section, we discuss related concepts and methods in graph partition.

Given a graph $G(V, E)$, a graph partition task is to partition all the vertices into several groups such that the connections between different groups are weak and the connections within the same group are strong. Many problems such as discovering communities in social networks can be formulated as a graph partition problem. For image segmentation, each pixel is considered as a vertex, and edge weights reflect the similarities between image locations. Many methods have been proposed for graph partitioning. In this thesis, we review three wildly-used ones: the min-cut method [Wu and Leahy, 1993], the normalized cuts method [Shi and Malik, 2000b] and the ratio-cut method [Wang and Siskind, 2003]. All of these methods extract information from the same *adjacency matrix* $W = (w_{ij})_{i,j=1...N}$. The weight w_{ij} is the weight of edge e_{ij} connecting the vertex i and vertex j. Note that we require W to be non-negative, i.e. $w_{ij} \geq 0$.

The degree of a vertex v_i is defined as:

$$d_i = \sum_{j=1...N} w_{ij} \tag{2.14}$$

The degree matrix D is a diagonal matrix in which $(D)_{ii} = d_i$. For a subset $A \in V$, the volume of A is denoted as:

$$\text{vol}(A) = \sum_{i \in A} d_i \tag{2.15}$$

In the following ,we focus on two-way graph partition.

2.3.1 The min-cut method

The most simple graph partition problem is min-cut [Wu and Leahy, 1993]. The cost of partition is called "cut". The cut between two vertex sets A and B is defined as:

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{2.16}$$

For two-way partition, we want to minimize the cut between a vertex set and the the rest of vertices:

$$\min_{A} \text{cut}(A, \bar{A}) \tag{2.17}$$

Where $\bar{A}$ denotes the complement of A in V. This objective function can be solved in polynomial time. However a major problem with min-cut is that the optimal solutions are usually isolated vertices, not meaningful for many applications.

2.3.2 The ratio-cut method

Considering the drawback of min-cut, ratio-cut optimizes a normalized objective function. For example, the ratio-cut method proposed by [Wang and Siskind, 2003] optimizes the following objective function:

$$\min_{A} \frac{c_1(A, \bar{A})}{c_2(A, \bar{A})} \tag{2.18}$$

where the function c_1 and c_2 are the cut measure (2.16) defined by weight matrices W_1 and W_2, respectively. The W_1 could be vertex affinities as usually; W_2 encodes various normalizing quantity, e.g. a constant function. In this case, c_2 is:

$$c_2 = \sum_{i \in A, j \in \bar{A}} 1 \tag{2.19}$$

The above objective function favors solutions with the minimal average cut. Like min-cut, ratio-cut can also be solved in polynomial time when the graph is planar [Wang and Siskind, 2003]. However, the bias towards unbalanced partition is not fully addressed.

2.3.3 The normalized cuts method

The normalized cuts method (Ncuts) optimizes an objective function normalized by both foreground and background volumes [Shi and Malik, 2000b]:

$$\frac{\text{cut}(A, \bar{A})}{\text{vol}(A)} + \frac{\text{cut}(A, \bar{A})}{\text{vol}(\bar{A})} \tag{2.20}$$

where $vol(A)$ is the volume of the set A.

With two ratio terms, this function favors balanced partitions. However, to solve this problem exactly is NP-hard. Approximate solutions can be obtained by spectral relaxation or semi-definite relaxation.

In the end, spectral relaxation comes down to finding the eigenvector with second smallest eigenvalue of the normalized graph Laplacian matrix L_s:

$$L_s = D^{-1}(D - W) \tag{2.21}$$

This problem is expensive to solve for large graphs with dense connections. Therefore, W is usually implemented as a sparse matrix. A drawback is that long range information is lost. As a remedy, the multiple-scale normalized cuts method exploits information in different scales [Cour et al., 2005]. Another method for fast eigenvector computation is based on the Nystrom approximation [Fowlkes et al., 2004]. The basic idea is to randomly sample a small submatrix of L_s and use its eigenvectors to approximate those of L_s.

2.4 Gaussian process latent variable model

Dimension reduction is a major research topic in machine learning. This section is about the GPLVM method which is used in Chapter 5 of this thesis [Lawrence, 2005]. GPLVM can be considered as non-linear generalization of the well-known PCA method. Due to its non-linearity, it can represent complex data variations in a latent space of small dimensionality. The Kernel PCA method (KPCA) can also be considered as non-linear generalization of PCA [Schölkopf, 2002]. The difference is that kernels in KPCA are applied to input features while kernels are applied to latent variables in GPLVM. The advantage of GPLVM over KPCA is two-fold. First, we can obtain a probabilistic distribution of features once the latent variables

are known. In contrast, KPCA does not admit a probabilistic interpretation. Secondly, the parameters of latent variables as well as the latent variables can be learned from training data. It is not possible for KPCA to do the same.

First, we will introduce some notations. Given N M-dimensional features $y_1, y_2, ..., y_N$, we can collectively represent them as a feature matrix $Y = [y_1, ..., y_N]$. An M-dimensional vector y^i denotes the i-th features of all samples. The q-dimensional vector x_i denotes the latent variable of sample i. All the latent variables are collectively represented as a matrix $X = [x_1, ..., x_N]$.

The basic assumption of GPLVM is that the distributions of feature dimensions are independent given the latent variables:

$$P(Y|X,\Theta) = \prod_{i=1}^{i=M} P(y^i|X,\Theta) \tag{2.22}$$

Where Θ denotes all parameters of the distribution. GPLVM also assumes that each feature dimension is from a Gaussian process parameterized by X and Θ.

$$P(y^i|X,\Theta) = N(y^i|0, K(X,\Theta)) \tag{2.23}$$

where the Gaussian distribution has zero mean and a covariance matrix $K(X,\Theta)$. This matrix is defined by a user-selected kernel function k as $K(X,\Theta)_{ij} = k(x_i, x_j, \Theta)$. As an example, $K(X,\Theta)$ could be XX^T. For clarity, the parameter Θ in the kernels is omitted.

Therefore the joint distribution of all features is:

$$P(Y|X,\Theta) = \frac{1}{(2\pi)^{MN/2}|K(X)|^{M/2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(K(X)^{-1}YY^T\right)\right) \tag{2.24}$$

Learning in GPLVM is to find the optimal X and Θ given the observed value Y. The standard maximal likelihood learning can be applied.

$$(X^*, \Theta^*) = \mathrm{argmax}_{X,\Theta} \quad P(Y|X,\Theta) \tag{2.25}$$

This will generally lead to a non-convex optimization problem, and can be solved by the stochastic gradient descend method. One major problem is that the computational cost increases cubically with the number of samples. To learn with a large amount of samples, fast approximation can be made based on the prior of sparsity [Lawrence, 2007].

2.4.1 Twin-kernel PCA

GPLVM can be further extended to twin-kernel PCA (TK-PCA). It replaces the inner products of features YY^T in Eq (2.24) by a kernalized matrix $K_Y(Y)$. Although the maximal likelihood learning algorithm can be applied without difficulty, the objective function can no longer be interpreted as likelihood. Therefore Lawrence et al. proposed that Eq (2.25) can be considered as minimizing the KL divergence of two Gaussian distributions with $K_Y(Y)$ and $K(X)$ as covariance matrices, respectively. Here, we introduce TK-PCA from a different perspective, i.e. as gaussian processes in a reproducing kernel Hilbert space.

According to Mercer's theorem, a positive semi-definite kernel can be decomposed into an inner product of two features, possibly of infinite dimensions.

$$k_Y(y_1, y_2) = \sum_i \lambda_i \psi^i(y_1)\psi^i(y_2) \tag{2.26}$$

Where λ_i and ψ are eigenvalues and eigenfunctions of the kernel k_Y, respectively.

Let $\Psi^i = (\psi(y_1), ..., \psi(y_N))$ denote the i-th features of all samples. As in GPLVM, we assume that each feature channel is independent, and is from a Gaussian distribution:

$$P(\psi^i|X, \Theta) = N(\psi^i|0, K(X, \Theta)) \tag{2.27}$$

Our key assumption is that the number of observations from channel i is proportional to its eigenvalue λ_i, and each observation is independent. Let $\bar{\Psi}^i$ denote λ_i times observation from this feature channel.

$$P(\bar{\Psi}^i|X, \Theta) = P(\Psi^i|X, \Theta)^{\lambda_i} \tag{2.28}$$

Then, the joint distribution of all feature channels are:

$$P(\bar{\Psi}^1, ..., \bar{\Psi}^N|X, \Theta) = \prod_{i=1}^{i=M} P(\bar{\Psi}^1|X, \Theta) \tag{2.29}$$

Replacing $P(\bar{\Psi}^1|X, \Theta)$ with $P(\Psi^i|X, \Theta)$ in Eq (2.26), we obtain:

$$P(\bar{\Psi}^1, ..., \bar{\Psi}^N|X, \Theta) = \frac{1}{(2\pi)^{\Lambda N/2}|K(X)|^{\Lambda/2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(K(X)^{-1}K_Y(Y)^T\right)\right) \tag{2.30}$$

where $\Lambda = \sum_i \lambda_i = \mathrm{tr}(K_Y(Y))$.

In maximal likelihood learning, the log likelihood of Eq (2.30) is maximized:

$$\ln P(\tilde{\Psi}^1, ..., \tilde{\Psi}^N | X, \Theta) = \text{Const.} - \frac{\Lambda}{2} \ln |K(X)| - \text{tr}\left(K(X)^{-1} K_Y(Y)^T\right) \tag{2.31}$$

Compared with the standard TK-PCA, this formulation has a coefficient Λ for $|K(X)|$. Alternatively, our interpretation uses a normalized kernel matrix $\frac{K_Y(Y)}{\text{tr}(K_Y(Y))}$, so that the latent space is invariant to the scale change of K_Y. Compared with the KL-distance based interpretation, this retains the probabilistic nature in GPLVM. The difference is that TK-PCA models the distribution of features in the *reproducing kernel Hilbert space* introduced by k_Y.

Chapter 3

A High-order Random Field for Contour Closure

3.1 Introduction

In the Oxford dictionary, a contour is defined as *an outline representing or bounding the shape or form of something*. The contour detection problem, as a fundamental vision problem, is to extract curves representing object shapes from images. Solving this problem can not only help visual systems to group pixels into objects but also reveal their semantic information [Palmer, 1999].

While many studies for contour detection focus on point-wise detection accuracy, our chapter with few others emphasize the topological properties of contours, in particular, contour closure. The *closure principle* has been observed by numerous psychological studies. In [Palmer, 1999], it is summarized as: "all else being equal, the elements which form closed figure tend to be grouped together". Using carefully designed stimuli, Kovacs and Julesz [Kovacs and Julesz, 1993] demonstrated that a set of contour fragments were more easily perceived from noisy background if they were closed. Therefore they remarked: *"a closed curve is much more than an incomplete one"*.

Aside from its psychological relevance, enforcing the closure principle could result in more meaningful contours. Due to noise and ambiguities, even state-of-the-art local edge detectors assign low probabilities to some of the true contour segments. Consequently, contours often break into fragments after thresholding. The bottom left image of Figure 3.1 shows a thresholded edge map of the Pb detector. It is in sharp contrast to the human labeling in the top right in terms of contour closure. This problem not only undermines the ability to recall all the edge points, but also causes difficulties to object and scene understanding. From structuralism's point of view, contours belonging to one object can be decomposed into several interconnected parts. When these structures are smeared in contour maps, it is difficult to learn

object representations in a bottom-up fashion.

Unlike other relatively-local Gestalt principles, the closure principle is "more global", making it not a simple task to enforce such a condition. We notice that, most contour grouping methods still tend to produce isolated or disconnected curve segments [Ren et al., 2008; Kokkinos, 2010b]. While some methods aim to enforce this principle [Wang et al., 2005; Mahamud et al., 2003], they often do so at individual contour level, and not are suitable for general natural images teemed with occlusion.

Recognizing this difficulty, Palmer [Palmer, 1999] suggested that one should complete the contours by performing image segmentation, e.g. by [Shi and Malik, 2000b]. Naturally, the boundaries of the segmented regions are closed contours. A recent work also enforced contour closure by an exponential number of constraints based on superpixel segmentation [Andres et al., 2011].

We consider as an interesting scientific question that whether closure can only be achieved with the help of region-based processing. If true, contours should be considered as by-products of image segmentation, and the contour detection problem ceases to be an independent problem. In addition, edge-based processing has its merits. For example, it has strong connection to the psychology [Palmer, 1999]. In addition, the junction properties of contours can be more conveniently modeled in a contour-based representation. Last, contours are usually sparsely distributed in images, and can lead to desirable sparse representations.

Our direction for this investigation is to construct a purely contour-based method which does not use segmentation information/representation at any level. If highly-completed contours can be achieved in the contour domain, it means that factors other than closure should be considered in order to differentiate region-based and contour-based processing.

To address the closure principle, our method first proposes two kinds of contour completion hypotheses which bridge the gaps between locally detected edges. Then a set of *contour connectedness* conditions are used to ensure that each edgelet must be connected to some of its neighbors. Although these connectedness conditions are only necessary rather than sufficient condition for the closure principle, this approximation leads to a more efficient model which often produces closed contours in practice. In addition to the closure principle, the principles of good continuity and proximity are also modeled by our energy function.

Collectively, our model is presented as a higher-order CRF model [Geman and Geman, 1984; Geman et al., 1990] based on a novel representation of image contours and their interactions. Our energy function includes unitary potentials reflecting local edge contrast, junction potentials encoding the continuity property, closure potentials enforcing correct contour topology, and a complexity potential which controls the level of details in outputs.

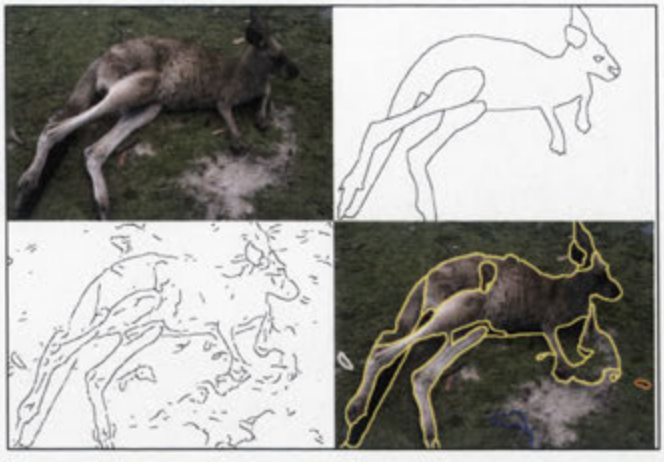


Figure 3.1: The goal of this chapter is to extract perceptually-salient and closed contours from images. **Top Left**: An image of a kangroo in the BSDS dataset. **Top Right**: A human-labeled contour image. **Bottom Left**: The contour map by Pb, as the input to our method [Martin et al., 2004]. **Bottom Right**: The contour map by our method.

The higher-order nature of our model originates from the connected conditions which take all the edgelets in a neighborhood into account. As shown in Figure 3.6, up to 25 edgelets may be involved in an energy term. In addition, it is shown in appendix A.1 that our energy function is generally non-submodular. To make inference efficient, we use the connectedness constraints to reduce some of the higher-order potential functions to linear order, which consequently enable us to formulate the CRF inference as an ILP (integer linear programming) problem. The rest of high-order potentials are represented by linear inequalities. An efficient algorithm is further devised to find a locally optimal integer solution of the ILP, taking advantage of the specialty of our ILP problem. In our experiments, solution energies are very close to lower bounds.

We have tested our method on both synthetic data and real images. Experiments show that it extracts multiple connected contours without loose ends, in accordance with our previous theoretical assessment. To show that our method is not completing contours in a native or blind way, our results are compared with several closely-related methods in terms of pixel-wise accuracy. On both the BSDS300 and BSDS500 datasets, our method is advantageous under the standard precision-recall metric and a newly developed metric for grouping error. Last but not least, the extracted connected contours appear clean and visually pleasing, perhaps suggesting that the results are closer to human's perception. In sum, our results show that the closure principle can be effectively enforced in contour domain. Further studies of factors other than closure are necessary for determining the computational nature of contour detection.

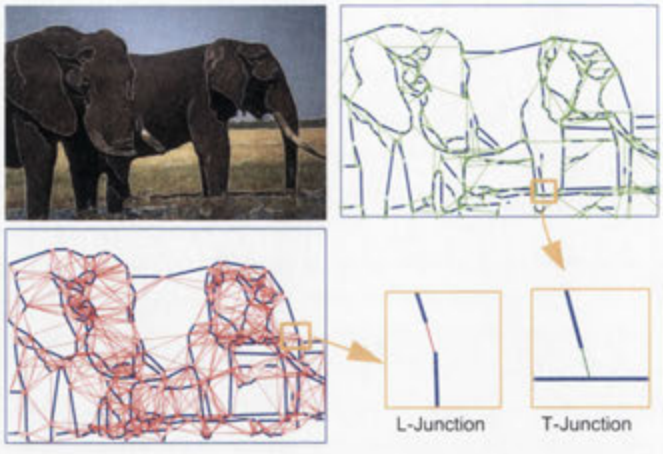


Figure 3.2: **Top left**: An input image overlayed with its thresholded Pb edges. **Top right**: Proposed T-junction completion edgelets are shown in green, and gradient edgelets by Pb are shown in blue. **Bottom left**: Proposed L-junction completion edgelets are shown in red. **Bottom right**: A zoomed-in view of the completion edgelets: L-junction(left) and T-junction(right). Best viewed in color.

3.1.1 Problem setup and chapter overview

Given a natural image, our method outputs a binary edge map indicating whether each pixel is an edge point or not. The amount of contours can be controlled by a parameter. A prominent feature of our results is that the contours are connected and there are no loose ends. Our method is purely contour-based, and does not use any segmentation. In Section 3.2, the contour completion problem is formulated as a probabilistic inference problem. Given local detection results, our model seeks the maximum a posteriori solution of the conditional probabilistic distribution of contours. The distribution encodes our prior about contours such as smoothness and connectedness. In Section-3.2, we describe how to construct our new CRF model, and how to use it to represent multiple image contours. Section-3.3 explains the CRF's potential function design. Section-3.4 is devoted to the proposed inference algorithm. The final two sections are experiments and conclusion.

3.2 Modeling multiple contours

3.2.1 The boundary segment graph

To facilitate contour completion, we need to design a proper representation that allows mid-, and long-range interactions among local edge elements. For this purpose, we first construct a graph of boundary segments based on the output of a local boundary detector, such as the Pb detector in [Martin et al., 2004]. On this graph, we then propose a higher-order CRF that

integrates local evidences with global constraints based on the Gestalt properties of contours at a scene level.

Our graph is built in two stages. We first form a set of short boundary segments using Pb, followed by a line fitting process. Specifically, we threshold Pb edge maps at the threshold of 0.05. Then, the binary edge map was linearized with Kovesi's line fitting algorithm [Kovesi]. This algorithm will break a line at extreme points. We refer to those line segments as *gradient edgelets* or G-edgelets in short. Normally, there are many gaps among G-edgelets, due to occlusions and miss-detections. So at the second stage, we introduce two new types of virtual *completion edgelets* (or C-edgelet in short) aiming to fill in the gaps and hence complete contours.

We first shorten each gradient edgelet by several pixels at its two endpoints such that it does not connect to other gradient edgelets. The first type of completion edgelets is proposed to link two neighboring gradient edgelets with the good-continuation property, referred as **L-junction** edgelets. Two gradient edges are considered to be in the same neighborhood if the distance between two of their endpoints are smaller than a threshold (say one fifth of the image width). L-junction edgelets connect gradient edgelets into longer contour segments by filling in the gaps caused by missing local cues and shortening. The second type of completion edgelets is used to capture the occlusion relationship between contours, referred as **T-junction** edgelets. A T-junction edgelet is placed between an endpoint of a gradient edgelet and another gradient edgelet if the extension of the former intersects the latter one without crossing other gradient edgelets. See Figure-3.2 for an illustration of gradient and completion edgelets. Note that for each endpoint, only one T-junction edgelet is proposed, and we treat four sides of image borders as gradient edgelets as they are also a kind of occluding (clipping) boundaries. Please note that there are important difference between L-junction edgelets and T-junction edgelets. An L-junction edgelet always bridges two endpoints of gradient edgelets whereas a T-junction edgelet connects one endpoint of an occluded edgelet to an interior point of an occluding edgelet.

To build a graph of boundary segments, we view each edgelet as a graph node. We use the same connectivity as we propose the completion edgelets. Note that in our graph the neighboring C-edgelets and G-edgelets always appear alternatingly. Compared with the CDT (Constrained Delaunay Triangulation) graph proposed in [Ren et al., 2008], our graph model accepts a higher order of connectivity, and it also explicitly encodes richer types of junction relations (such as occluding/occluded) among contours. Moreover, we have experimentally confirmed that the proposed graph with about 1000 completion edges per image is able to recall 95% of the ground-truth boundary points on the BSDS300 dataset. In comparison, a

CDT graph with about 300 completions per image has a recall rate of 88%. Although a CDT graph seems more efficient, the high recall rate of our model is a precondition for enforcing the connectedness constraints. At the same time, the model adopt a fully factorized representation of the junction potentials, so that the model complexity only increases moderately with the number of completion edges.

3.2.2 A higher-order CRF for contours

Based on the graph of boundary segments in Section 3.2.1, we build a CRF model to capture both properties of individual contours, and their interactions. In particular, our model focuses on four aspects of contour properties, including 1) local image contrast; 2) contour smoothness and continuity; 3) contour closure; and 4) overall model-complexity.

Our model is formulated as follows. Let an image I have a boundary segment graph $G = (V, E)$ with edgelets as its node set V. We associate a binary variable with each edgelet, and denote all variables as $\mathbf{Y} = \{y_i \in \{0,1\}, \forall i \in V\}$. The edgelet variables are divided into two sets: the gradient edgelets $\mathbf{Y}^g = \{y_i, i \in V_g\}$ and the completion edgelets $\mathbf{Y}^c = \{y_i, i \in V_c\}$. When we need to differentiate two types of completion edgelets, we use V_c^l and V_c^t to denote the sets of L-junction edgelets and T-junction edgelets, respectively. Each edgelet is connected to neighboring edgelets at the vicinity of its two endpoints. To capture the "good continuity" and "closure" rules, we need to consider both near-range and longer-range interactions between the edgelets, by introducing the following three types of cliques in the graph:

1. For every connected edgelet pair, we have a "pairwise clique", and denote the set of pairwise cliques as C^p;
2. We assign every completion edgelet (no matter it is L-type or T-type) a triple-node clique that includes itself and two neighboring gradient edgelets, and this type of clique is called "junction clique", and the entire set of such junction-cliques is denoted as C^l;
3. At an endpoint of a given gradient edgelet, all the edgelets connecting to this endpoint induce a higher-order clique, and we refer to the set of such higher-order cliques as C^H.

Our CRF defines a joint distribution of the labels $\mathbf{Y}$ given the input observation $\mathbf{X}$, denoted

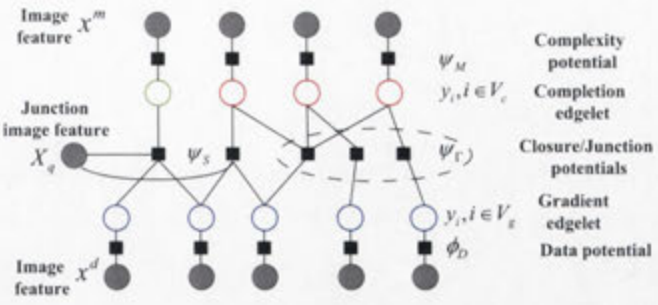


Figure 3.3: A factor graph representation of our CRF model. The circles represent variables in the model and the squares are potential functions (i.e. factors). The gradient edgelets are shown in blue and the completion edgelets are in red or green. Some connections are not shown for clarity and see text for details.

as $P_{\mathbf{Y}|\mathbf{X}}$ for short, which includes four types of potential functions,

$$P_{\mathbf{Y}|\mathbf{X}} = \frac{1}{Z_{\mathbf{X}}} \exp\left\{ -\left(\sum_{i \in V_g} \phi_D\left(y_i, \mathbf{x}_i^d\right) + \sum_{q \in C^I} \psi_I\left(\mathbf{Y}_q, \mathbf{X}_q\right) \right.\right.$$
$$\left.\left. + \sum_{q \in C^P \cup C^H} \psi_\Gamma\left(\mathbf{Y}_q\right) + \sum_{i \in V_c} \psi_M\left(y_i, \mathbf{x}_i^m\right) \right) \right\}, \tag{3.1}$$

where $Z_{\mathbf{X}}$ is the partition function, and the potential functions ϕ_D (data term), ψ_I (junction term), ψ_Γ (global closure/connectedness effect term) and ψ_M (model-complexity term) are used to describe different aspects of desirable contour properties. We use $\mathbf{Y}_q$ to represent the variables associated with clique q.

A *factor graph* representation of the CRF model is shown in Figure 3.3. The circles represent the variables in the model and the squares are potential functions. The black circles in the uppermost layer denote local observations. In the layer below, the green circles stand for the binary random variables of gradient edges, while the purple circles in the bottom layer stand for the binary labels of completion edgelets. The connection between a gradient edge node and an observation node represents a unary term in our potential function, encoding the information from local edge detection.The square nodes between gradient edge nodes and completion edge nodes represent junction potential functions. The prior terms which are unary terms are not shown in this graph for clarity.

3.3 Design of potential functions

3.3.1 Unary data terms ϕ_D

A unary potential $\phi_D\left(y_i, \mathbf{x}_i^d\right)$ computes the likelihood score that the i-th edgelet lies on a true contour. This term is defined as a linear function of a boundary feature vector $\mathbf{x}_i^d$:

$$\phi_D\left(y_i, \mathbf{x}_i^d\right) = \alpha \mathbf{w}_d^T \mathbf{x}_i^d y_i, \tag{3.2}$$

where $\mathbf{w}_d$ is the weight for image features and α is an overall weight for the data term. In this work, we use the lengths of edgelet l_i, logarithm of the average Pb value, and their product as input features.

3.3.2 Junction potentials ψ_J

For every completion edgelet $y_i, i \in V_c$ and the associated triple-node clique $q \in C^J$, we define a junction potential $\psi_J(\mathbf{Y}_q, \mathbf{X}_q)$ to encode the continuity property. For an L-junction, we design an L-potential to impose the principle of good-continuity; For a T-junction, we design a T-potential to express the likelihood of occluding/occluded relationships. For either case, we define an image-dependent triplet potential function involving the central completion edgelet y_i and its two connected neighbors $\{y_j, y_k\}$, as shown below (and in Fig-3.4):

$$\psi_J(\mathbf{Y}_q, \mathbf{X}_q) = \psi_J(y_i, y_j, y_k, \mathbf{X}_q) = \mathbf{w}_J^T \mathbf{X}_q y_i y_j y_k, \tag{3.3}$$

where $\mathbf{w}_J$ is the coefficient for the junction feature vector $\mathbf{X}_q$, which is extracted from the neighborhood of clique q. Note that this potential assigns a score of $\mathbf{w}_J^T \mathbf{X}_q$ to the case that the whole triplet is active, and zero otherwise.

The image feature vectors used in this work are summarized in Table-3.1 (for notation please refer to Figure-3.4). For the L-junction case, we denote the features by $\mathbf{X}_q^l$, and for the T-junction case we use $\mathbf{X}_q^t$. The first feature of $\mathbf{X}_q^l$ is l_i denoting the distance between the two endpoints. The second feature $\tilde{l}_i$ is the sum of two distances between each endpoint and the estimated center of the junction. The angular completion features are adopted from [Sharon et al., 2000]. These feature measures the turning angle of the junction, and the difference of two angles at each end of the completion edgelet. According to [Sharon et al., 2000] the linear combination of these two features is a scale invariant approximation of the elastica energy. For T-junction edgelets, the first feature l_i denotes the edge length, the second feature is the ratio between the length of the T-junction edgelet and the length of the occluded gradient edgelet. The third feature measures the deviation of the junction center from the center of the occluding

	image features	description
$\mathbf{x}_q^l(1)$	l_i	effective distance(smooth)
$\mathbf{x}_q^l(2)$	$\tilde{l}_i$	effective distance (corner)
$\mathbf{x}_q^l(3)$	$a_i(\theta_{i,j}+\theta_{i,k})^2$	angular completion
$\mathbf{x}_q^l(4)$	$b_i(\theta_{i,j}-\theta_{i,k})^2$	angular completion
$\mathbf{x}_q^t(1)$	l_i	effective distance
$\mathbf{x}_q^t(2)$	l_i/l_j	relative distance
$\mathbf{x}_q^t(3)$	$\frac{\min(l_k^a,l_k^b)}{l_k^a+l_k^b}$	intersection position
$\mathbf{x}_q^t(4)$	$(\theta_{i,k}-\pi/2)^2$	intersection angle
$\mathbf{x}_i^d(1)$	l_i	length of edgelets
$\mathbf{x}_i^d(2)$	$\log(Pb_i)$	average log(Pb) response
$\mathbf{x}_i^d(3)$	$l_i\log(Pb_i)$	sum of log(Pb) response

Table 3.1: Summary of image features used in junction potential functions (c.f. Fig-3.4). For completeness, we also include the unary data term features $\mathbf{x}_i^d$.

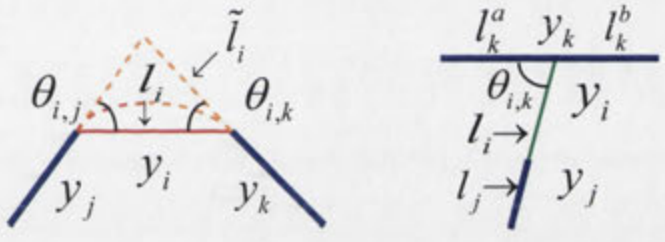


Figure 3.4: Two types of completion edgelets. **Left**: an L-junction edgelet and its image feature description; **Right**: a T-junction edgelet and its image feature description.

edgelet. The last feature is the squared difference between $\pi/2$ and the angle of this T-junction. 6

3.3.3 Contour closure potentials ψ_Γ

The contour closure principle is also difficult to capture by local potential functions. In this chapter we approximate the global closure principle by a sequence of *connectedness constraints*. At an endpoint of an edgelet we can identify two types of such connectedness constraints, each of them is formulated as a set of linear inequalities given follows:

(1) Completion Constraints ensure that no completion edgelet can be active without both of its connected gradient edgelets being active. That is, at either endpoint of a completion edgelet $y_i, i \in V_c$, its neighboring gradient edgelet $y_j, j \in V_g$ should satisfy the inequality,

$$y_i \leq y_j, \quad \forall i \in V_c, j \in V_g, (i,j) \in C^p. \tag{3.4}$$

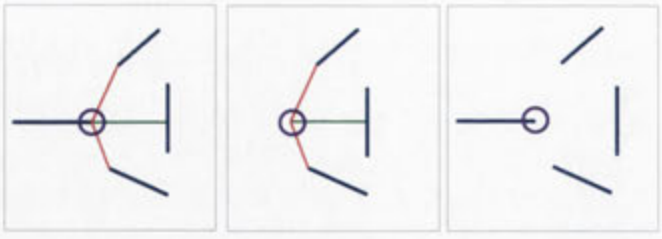


Figure 3.5: Examples of valid/invalid configurations w.r.t. the contour closure potential. Blue: Gradient edgelets; Red/Green: Completion edgelets. **Left**: A valid configuration which satisfies the closure potential. **Middle**: A configuration which violates the completion constraint (i.e. Eq-(3.4)). **Right**: A configuration which violates the extension constraint (i.e. Eq-(3.5)).

(2) Extension Constraints ensure that if a gradient edgelet is active, then at least one of its connected completion edgelets should be active such that it can be extended. Formally, at either endpoint of a gradient edgelet $y_j, j \in V_g$, all the edgelets incident to that endpoint, which form a clique $q \in C^H$, should satisfy the inequality,

$$y_j \leq \sum_{i \in q \cap V_c} y_i, \quad \forall j \in V_g, q \in C^H \tag{3.5}$$

Figure-3.5 illustrates some configurations that either satisfy all the above constraints, or violate one of these constraints. Together, inequalities-(3.4) and -(3.5) ensure the connectedness of contours, and hence in the solution space, no contour will be extracted with loose ends. Each of the above inequality constraints can be coded as a polynomial term which equals to 0 when the inequity is satisfied, and equals to $\mathtt{M}$ which is a sufficiently large number, otherwise. The inequity-(3.4) is coded as $\mathtt{M}(1-y_j)y_i$, and the inequity (3.5) is coded as $\mathtt{M}y_j \prod_{i \in q \cap V_c}(1-y_i)$. We collectively represent all the connectedness constraints by the contour closure potentials, $\sum_{q \in C^P \cup C^H} \psi_\Gamma(\mathbf{Y}_q) =$

$$\mathtt{M}\Big(\sum_{\substack{(i,j) \in C^P \\ i \in V_c, j \in V_g}} (1-y_j)y_i + \sum_{\substack{q \in C^H \\ j \in V_g}} y_j \prod_{i \in q \cap V_c} (1-y_i) \Big),$$

where $\mathtt{M}$ is a very big positive number. Note that $\psi_\Gamma(\mathbf{Y}_q) = 0$, if and only if the corresponding inequality is satisfied.

3.3.4 Model complexity potentials ψ_M

In natural images, image contours often have multiple levels of details, depending on the scale at which a scene is perceived. To reflect this, we introduce a fourth-type potential function, which can be viewed as adding a preference towards reducing the total effective length of

completion edgelets, hence controls the overall model complexity:

$$\psi_M(y_i, \mathbf{x}_i^m) = \tau \mathbf{w}_m^T \mathbf{x}_i^m y_i, \tag{3.6}$$

where $\mathbf{x}_i^m$ is a feature representing the effective length of the completion edgelet y_i, $\mathbf{w}_m$ the (negative valued) weighting coefficients, and τ is a user-specified global scalar controlling the overall model complexity. Details will be given in the following sections. Note that this term is simply a weighted "label cost" used in Delong et al. [2012], which is a global higher-order term.

3.3.5 Energy function simplification

The high order terms in (3.3) make the energy minimization problem difficult to solve. Fortunately, due to the speciality of our potential functions, especially noting that the y_is are boolean variables, we realize that when Eq-(3.4) is true, then the cubic-term triplet junction potential in Eq. (3.3) can be reduced to linear terms, i.e.

$$\mathbf{w}_J^T \mathbf{X}_q y_i y_j y_k = \mathbf{w}_J^T \mathbf{X}_q y_i, \quad \forall i \in (q \cap V_c),\ q \in C^J. \tag{3.7}$$

This reduction is significant, as it greatly simplifies our energy function. Now, except for the higher-order terms of ψ_Γ, all the other terms are reduced to be unary and linear. Moreover, even the higher-order terms ψ_Γ are in *linear* form, because they are nothing but the linear inequalities in Eq-(3.4) and Eq-(3.5).

In summary, given input $\mathbf{X}$, the problem of maximizing the conditional probability (3.1) can be transformed into an Integer Linear Program (ILP):

$$\begin{aligned} &\min_{\mathbf{Y}} \alpha \sum_{i=1}^{N} \mathbf{w}_d^T \mathbf{x}_i^d y_i + \sum_{q \in C^J \wedge i \in q \cap V_c} \mathbf{w}_s^T \mathbf{X}_q y_i - \tau \sum_{i \in V_c} \mathbf{x}_i^m y_i \\ &s.t. \qquad \psi_\Gamma\left(\mathbf{Y}_q\right) = 0, \quad \forall q \in C^P \cup C^H. \end{aligned} \tag{3.8}$$

The constraints are simply the linear inequalities in Equation (3.4) and (3.5).

3.4 Inference

After combining all potential functions together, we obtain a (very) higher-order CRF model. To directly solve this higher-order CRF is very difficult. First of all, unlike previous CRF models, our graph construction allows much larger cliques to form. Therefore, it is non-trivial to

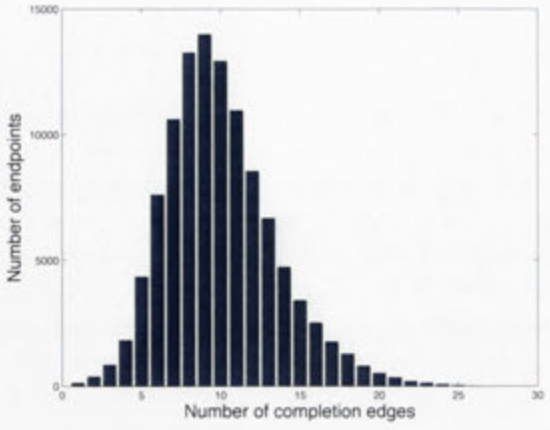


Figure 3.6: The histogram of the number of completion edgelets connected to one endpoint on the BSDS dataset. We can see that there are some junctions with dense connections.

apply message-passing algorithms, such as Loopy BP, to the problem. Secondly, it is shown in Section 3.4.1 that the higher-order inequality constraints in ψ_Γ make the whole energy function *non-submodular* under very mind conditions. This non-submodularity and high-order properties of our energy function prove difficult to apply graphcut [Boykov et al., 2001] or QPBO [Hammer et al., 1984] for inference.

Therefore, a customized inference algorithm based on linear relaxation is proposed in Section 3.4.2. The feasibility and efficiency of this algorithm is discussed in Section 3.4.3 and Section 3.5.3.

3.4.1 Properties of the proposed energy function

First, unlike previous boundary CRF models, our graph construction allows large-sized cliques to form. As shown in Figure 3.6, the mean value of vertex degrees is around 10 and the maximum is 25 for 100 BSDS images. Secondly, although most part of the energy function is linear, the higher-order constraints in ψ_Γ make the whole energy function non-submodular. It can be shown that if there exist two single-connected contours which share some edgelets, then the energy function is non-submodular. See the appendix A.1 for detailed proof.

Proposition 1. *If there are at least two single-connected contours intersecting with each other, then the energy function in Eq.*(3.1) *is non-submodular.*

Empirically we also observe that the assumption is satisfied in most cases. In the following subsection, we propose a novel optimization method based on the cutting-plane and coordinate descent methods.

3.4.2 Algorithm description

For clarity of presentation, the energy minimization problem is formulated as a standard integer program:

$$\min_{Y} \quad c^T \mathbf{Y} \tag{3.9}$$

$$s.t. \quad A\mathbf{Y} \leq b \tag{3.10}$$

$$y_i \in \{0,1\}, \forall i \in V, \tag{3.11}$$

where $\mathbf{Y} = \{y_i, i \in V\}$ is the labels of all edgelets. The parameter vector c is the weights of edgelets, and A, b parameterize linear constraints for the connectedness constraints.

While there exist many general-purpose off-the-shelf ILP solvers, such as those based on branch-and-bound, they are *incapable* to handle large-scale problems such as our case (where there typically are thousands of variables and constraints to be solved). We therefore propose a tailored optimization approach, which combines the ideas of cutting-plane and coordinate-descent.

Our inference algorithm is shown as Algorithm 1. We firstly solve a *linear relaxation* of the original integer program. Usually, after each round some variables have fractional values and we sequentially add more constraints such that those fractional variables have to be boolean. Unlike the conventional cutting-plane method, we directly add the integral constraints to a small number of fractional variables instead of searching for a cut. In particular, we randomly select N_{max} fractional variables, and add their integer constraints into Eq. (3.8). Then we solve a mixed integer linear programming (MILP) that generates integer solutions to the selected fractional variables. Once a solution to the MILP is found, we fix the values of the selected subset of variables and search an optimal configuration for the remaining variables iteratively.

Since this algorithm is not guaranteed to find global optimum, it is necessary to assess the quality of solutions. Section 3.4.3 first shows that the algorithm can always find a feasible solution to the integer program. Then Section 3.5.3 shows that the solutions are also close to the optimal, and can be computed efficiently for our problem.

3.4.3 Feasibility of Algorithm 1

A solution of our algorithm always satisfies connectedness constraints since all the constraints are present in the inference process. In other words, the solution will correspond to topologically correct contours. However, it is not self-evident why the algorithm can always produce a feasible solution. Note that some of the labels are greedily determined when others still

Algorithm 1 Our inference method for solving (3.8)

Input: A, b, c, N_{max}

$D_{int} \leftarrow \{\emptyset\}$

$D_{sol} \leftarrow \{\emptyset\}$

$$\mathbf{Y}^* = \arg\min c^T \mathbf{Y} \tag{3.12}$$

$$s.t.\ A\mathbf{Y} \leq b \tag{3.13}$$

$$0 \leq y_i \leq 1, \forall i \in V \tag{3.14}$$

$N_f \leftarrow$ number of labels with fractional values in $\mathbf{Y}^*$

while $N_f > 0$ **do**

  if $N_f < N_{max}$ **then**

    $D_{int} \leftarrow$ indices of N_f fractional labels in $\mathbf{Y}^*$

  else

    $D_{int} \leftarrow$ indices of N_{max} fractional labels in $\mathbf{Y}^*$

  end if

$$\mathbf{Y}^* = \arg\min\ c^T \mathbf{Y} \tag{3.15}$$

$$s.t.\ A\mathbf{Y} \leq b \tag{3.16}$$

$$y_i \in \{0,1\}, \forall i \in D_{int} \tag{3.17}$$

$$y_i = y_i^{sol}, \forall i \in D_{sol} \tag{3.18}$$

$$0 \leq y_i \leq 1, \forall i \in V \tag{3.19}$$

  $\mathbf{Y}_{sol} \leftarrow \mathbf{Y}^*$

  $D_{sol} \leftarrow D_{sol} \bigcup D_{int}$

  $N_f \leftarrow$ number of labels with fractional values in $\mathbf{Y}^*$

end while

return $\mathbf{Y}^*$.

have fractional values. Therefore it seems that there could be an infeasible MILP such that Algorithm 1 cannot find a solution at all. The following proposition excludes this possibility:

Proposition 2. *Algorithm 1 can always find a feasible solution of Eq*(3.9)-(3.11).

The proof is shown in the appendix A.2.

3.5 Experiments

Learning model parameters. The parameters used in the our model, i.e. $\mathbf{w}_d, \mathbf{w}_l, \mathbf{w}_m$ (cf. Eq-(3.8)), are learned by the logistic regression with a piece-wise learning strategy [Sutton and McCallum, 2009]. For each type of edges, we labeled hundreds of negative and positive samples. The logistic regression is then used to determine the optimal parameters. Cross-validation is used to choose the global parameter α in unary terms (Eq-(3.8)).

3.5.1 Tests on synthetic images

In order to validate our contour completion model, we test it on a number of purposely designed synthetic images. These images are commonly used in cognitive vision.

Figure-6.2 shows some examples where our method performs nearly perfectly, extracting multiple closed contours regardless of occlusion (1st image), clutter (2nd image), and prefers closed contours (3rd image), just as we expect. In this figure, we give the raw outputs of our method, where different colors are used to indicate different types of edges: blue stands for gradient edgelets (G-edge); red stands for L-junction edgelets (L-edge); green stands for T-junction edgelets (T-edge). Note that the third image is often used for demonstrating Kanizsa's visual illusionary contours. Our model shows an evident preference of closed contours, similar to human's perception. The L-junction edgelets in-between are turned "on" because they lead to a lower cost to the energy function.

3.5.2 Tests on natural images

Example results. We test our model on BSDS300, and also some other natural images (such as the Weizmann horse dataset Borenstein and Ullman [2002a] etc). Satisfactory results are obtained. Figure-3.9 gives some sample results of our method, with both raw outputs and the final contour maps displayed. More contour maps overlaid on BSDS300 images can be found in Figure-3.8. It seems that our method produces very clean, and also connected contours. Even though each of these contours does not necessarily correspond to a semantically meaningful surface region (which would require a higher level of vision processing such as

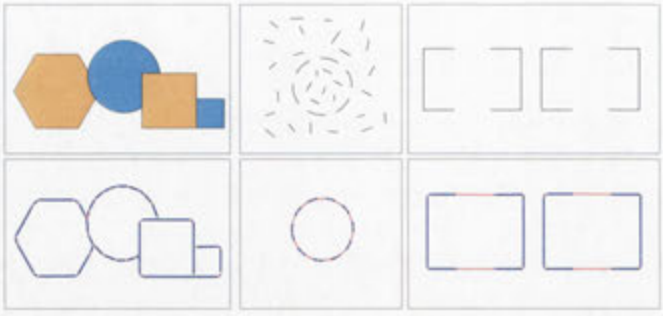


Figure 3.7: **Top row**: three synthetic test images (from left to right: occlusion, clutter, closure). **Bottom row**: Our results. The lines in blue indicate the active gradient-edges; The lines in red indicate the active L-junction edgelets, and the lines in green indicate the active T-junction edgelets.

figure-background segmentation), our results do improve over Pb results significantly, and will be helpful for later-stage high-level vision processing. Our `Matlab` implementation running on a 2.1 GHz Intel Core Duo CPU, takes about 5 minutes to process a BSDS image (excluding the time used for Pb detection).

Model complexity. In this experiment, we aim to test the effect of tuning our model-complexity parameter τ. By tuning this parameter, we obtain a series of results, each with a different level of details (complexity). Some examples are shown in Figure-3.10. Clearly, when τ is larger, the extracted contour images will have less details, and *vice versa*. It is worth noting that, in both cases, the connectedness of contours is always maintained, thanks to our hard closure constraints used in the inference. This is in sharp contrast to local methods such as Pb, for which increasing their threshold tends to yield more-fragmented contours.

The closure potentials. This chapter's key insight is about the closure principle. In this experiment, we deliberately exclude the two sets of inequality constraints and run inference again.Without the inequities, the probability of each edge is determined solely by its weight.

We have tested two cases, one with the closure potentials, and one without, on all the 300 images in BSDS300. Figure-3.11 gives a statistical comparison using the precision-recall curves. From this figure, the effects of the closure potentials are obvious.

3.5.3 Effectiveness of the inference algorithm

We demonstrate the effectiveness of the inference algorithm on 100 BSDS images. The quality of a solution is estimated by computing the ratio of solution energy over the energy lower bound obtained by linear programming relaxation. Since in our model energies have negative values, the ratio is a number between zero and one. The histogram of energy ratios is shown in the left half of Figure 3.12. It shows that the energy of our solutions are very close to the lower

Figure 3.8: Contours overlaid with the input images from the BSDS300 dataset. Best viewed in color.

Figure 3.9: Sample results of our method on natural scene images. For every three rows, **top row**: the input images; **middle row**: our method's raw outputs (blue: G-edge; red: L-edge; green: T-edge); **bottom row**: Our method's final contour maps. (**Better viewed on screen with zoom-in**).

Figure 3.10: Effect of tuning the model complexity parameter τ. **Row 1**: The input images. **Row 2∼5**: Our method's outputs when $\tau = 0, 1, 2, 3$. As τ increases, the extracted contour images contain less details, yet connectedness is well maintained.

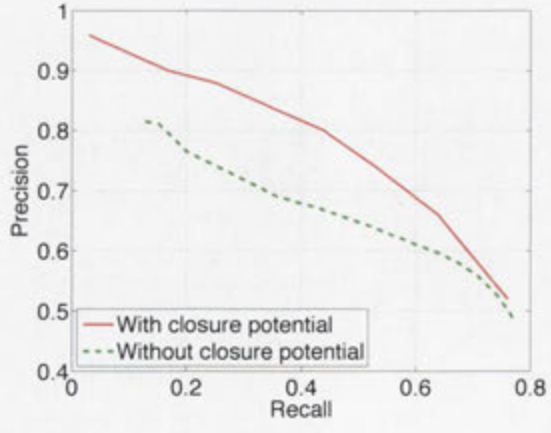


Figure 3.11: Effect from the closure potentials. It is clear that the closure-potentials substantially boost model's overall performance.

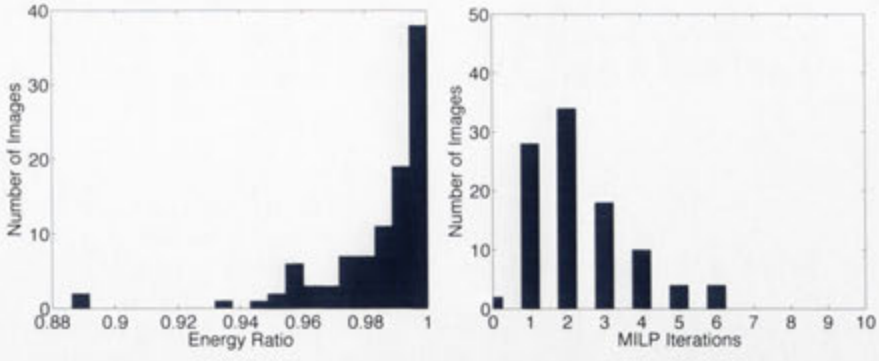


Figure 3.12: The left figure shows the histogram of the ratios of solution energy over lower bound energy. The right figure shows the histogram of MILP iterations per image. The experiment is conducted on 100 BSDS images.

bounds. The average ratio is 0.982. Second, we show that the number of MILP iterations is far less than the theoretical upper bound $\left\lceil \frac{N}{N_{max}} \right\rceil$. As shown in the right half of Figure 3.12, our inference algorithm only needs to solve up to 6 MILP to obtain a solution while the upper bound is around 100 iterations. This is not surprising as variables are coupled by the linear constraints. When one set of variables are fixed to integers, many other variables will have integer value due to these constraints.

We also tried the multi-start procedure to further minimize the solution energy. The procedure selects ten different sets of the variables for branch in Algorithm 1. Therefore, ten solutions will be obtained by the procedure. The solution with the minimal energy is selected as the final solution. The average energy ratio increased to 0.994. Although the energy is a little more close to global optimal, the contour outputs are little different from those obtained by Algorithm 1. Therefore, we think the Algorithm 1 is enough to obtain a good solution, and the multi-start procedure is not employed for the rest of the experiments.

3.5.4 Benchmark with existing methods

We compared our method with other existing methods for contour extraction and completion. Some example results are shown in Figure-3.13 for visual evaluation. Compared with Pb, Ren et al.'s CRF and the contour-cut algorithm, our method seems to produce better results in terms of contour connectedness. Of course, such a comparison may be seen unfair, as it is not other algorithms' intention to generate connected contours. Nevertheless, we find our results visually more pleasing, which perhaps suggest that our results are closer to human perception. We also did an overall statistical comparison among these algorithms, based on BSDS300 benchmark. The precision-recall curves are plotted in Figure-3.14. Our new model outperforms Ren et al.'s

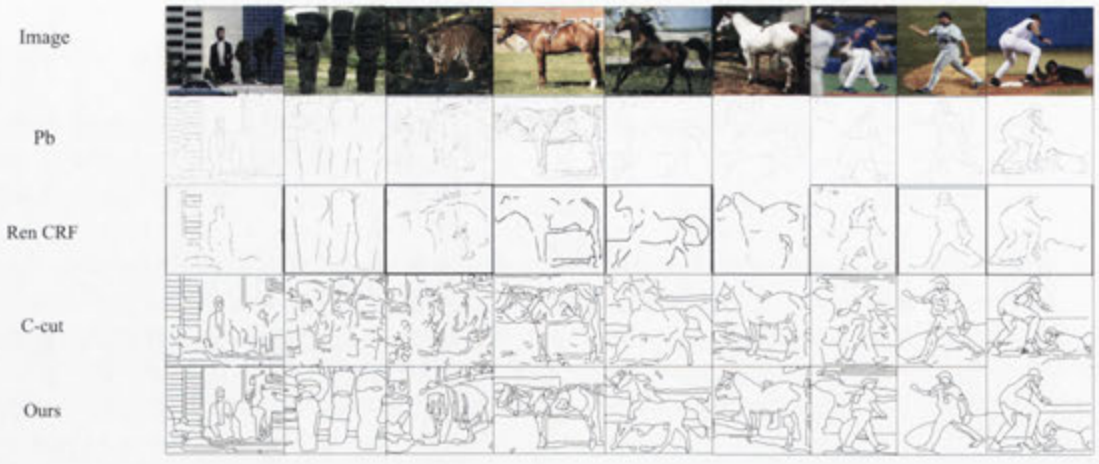


Figure 3.13: Methods comparisons on natural scene images: **Top row**: sample images from the BSDS dataset [Arbelaez et al., 2011], the Weizmann horse dataset Borenstein and Ullman [2002a], and baseball player datasetMori et al. [2004]. **Other rows** (from top to bottom): the Pb detector, Ren's CRF (reproduced from Ren et al. [2008]), the contour-cut method, and our method ($\tau = 0.5$).

CRF model by a clear margin. In the high-precision regime, our model achieves comparable result with the contour-cut algorithm (using its top 10 contours only), but in the high-recall regime (this regime is most useful for object recognition Kokkinos [2010b]) our method's performance is more consistent. We did not include Kokkinos' method Kokkinos [2010b] in the comparison, because he used a dedicated local edge detector (instead of Pb).

Our method is further tested on the BSDS500 dataset which has additional 200 test images. As shown in Figure 3.15 the general trends of the results are similar to those in BSDS300. Our model achieves the same F-value 0.67 as the Pb detection. However, our model is advantageous when the recall rate is neither too high nor too low. This is understandable since the priors of our model, i.e. smoothness, connectedness are most effective in the mid-recall range. When the recall rate is high, the contours obtained from thresholding Pb detection have very small gaps. Therefore, even when our model connects the contours correctly, the recall rate will not benefit much. The incorrect completions however, will lower the precision. In the low recall, high precision region, salient gradient edges are sparse, and it becomes more difficult for our model to connect these sparse edges. According to Kennedy et al. [2011], contours associated with first 20 eigenvalues are selected as outputs. Their performance is enhanced by spectral information in the high precision region.

Contour Rand index. As discussed in Kokkinos [2010b]; Kennedy et al. [2011]; Felzenszwalb and McAllester [2006], the precision-recall metric does not take grouping error into account. To measure grouping performance, we adapt the well-known *Rand Index* metric Rand

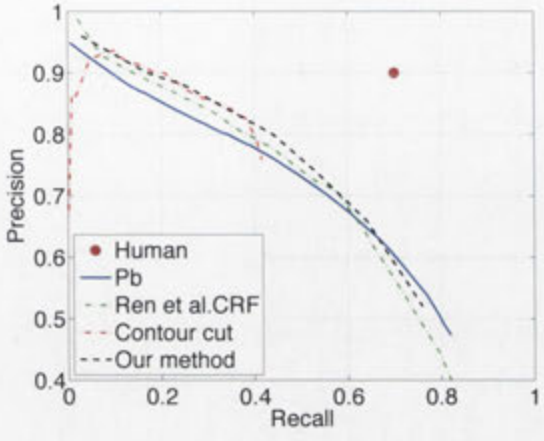


Figure 3.14: Precision-Recall curves for 4 methods on the BSDS300 dataset (Better viewed on screen).

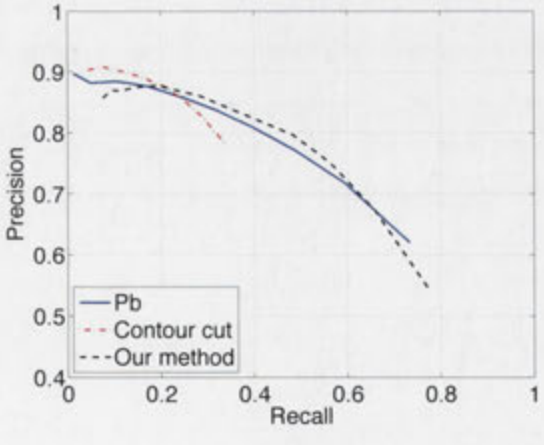


Figure 3.15: Precision-Recall curves on the BSDS500 dataset (Better viewed on screen).

[1971] to the contour case –we call our new metric the Contour Rand Index (CRI). The Rand index is a classic metric for grouping error and defined as follows: Let $S = \{e_1, e_2...e_n\}$ be a set of elements. $C = \{c_1, c_2...c_n\}$ and $\tilde{C} = \{\tilde{c}_1, \tilde{c}_2...\tilde{c}_n\}$ are two sets of group IDs for every element. Two elements are in the group if they have the same group ID. Let a be the number of element pairs that are grouped together according to either C, or $\tilde{C}$. b be the number of pair of elements in different groups according to either C, or $\tilde{C}$. p be the number of all pairs of elements. Rand index is

$$RI = \frac{a+b}{p} \tag{3.20}$$

To adapt the Rand index to the contour grouping problem, we make use of the groundtruth contours in the BSDS dataset. The objective is to compute Rand index as the similarity of grouping between a contour image and human labeling. However, two problems need to be solved. First we need to determine the elements for grouping. Rand index is define on the same set of elements. Apparently, a groundtruth image and a contour image have different sets of contour points. Second, we need to determine which elements are in the same group, given a contour or groundtruth image. To address the first problem, the bipartite matching is carried out between the groundtruth and contour image. Every matched pair of points is considered as one element in Rand index. To address the second problem, we group contours based on the connectivity. We search for 8-connected contour points in a contour image as a group, and assign these connected points a common group number. Searching stops when all the contour points have a group number. The same process is carried out for groundtruth. A pair of points with same group number in the test image is correctly grouped if their matched points in the groundtruth have the same grouping number. If these two points are assigned with different group numbers in the test image, their counterparts in the groundtruth need to have different group numbers. To improve the stability of the metric, we remove the contour groups containing less than 10 image pixels. Note that although all the contour points have a group number, only matched points are taken into account by the metric. Last, the Rand index is averaged across the images and human subjects. In sum, CRI is defined as:

$$CRI = \frac{\sum_i \sum_j (a_{i,j} + b_{i,j})}{\sum_i \sum_j p_{i,j}}, \tag{3.21}$$

where i is the index of human subjects and j is the index of test images. The result is given in Figure-3.17, which shows that our method is much better than GPb, and is very close to human's performance.

The CRI of our model and other models are shown in Figure 3.17. It shows that our model

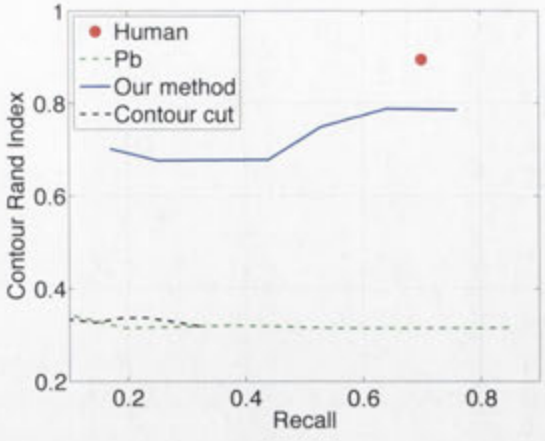


Figure 3.16: Performance comparison by Contour Rand Index. Human CRI is shown as a red dot.

consistently outperforms Pb in all recall region. Our model's outputs also outperform the contour-cut algorithm.

3.6 Junction edge validation

In our model, linearized contours are connected by L-junction edgelets and T-junction edgelets. While they are mainly used for connecting edge fragments, it is interesting to see that whether true junctions perceived by human can be obtained from these junction edges. After obtaining a solution from the ILP, all the completion edgelets with label "1" are treated as a potential junction. To predict its probability, several features are taken into account such as the turning angle, the pb value and the length of adjacent gradient edges. Then, these features are fitted into a logistic classifier, and the prediction of the classifier is considered as the probability of the junction. For one image, multiple contour solutions are produced with several different complexity parameters. To combine all these output, a second logistic classifier is build to make the final prediction based on the probability values in different solutions. Some outputs of our algorithm are shown in Figure 3.17. Only junctions with probability larger than 0.5 are displayed in circles, and the probability of a junction is coded by color. This figure shows that our outputs largely correspond to true image junctions.

To benchmark the performance of the junction prediction method, we use the groundtruth and benchmark algorithm introduced in [Maire et al., 2008]. The precision-recall curve of our outputs is shown in Figure 3.18, along with those from the contour-based probabilistic junction method (Pj for short) [Maire et al., 2008], and the Harris corner detector [Harris and Stephens, 1988] (on Pb result). It shows that our model's performance is between the Harris detector and

Figure 3.17: Sample results of junction detection on BSDS300 dataset. The first and third column show the detected junctions in the original image. The second and forth row show the junctions overlaid on the Pb results. The junctions with darker color has higher probability. Better viewed in color.

Pj. Pj (F-value 0.38) outperforms our method (F-value 0.37). This is understandable since our method is neither designed nor trained to detect junctions optimally. For example, a Y-junction is factorized into several L-junctions in our model to make inference efficient in spite that Y-junctions have different statistics than L-junctions. Nevertheless, our performance is still better than the Harris detector (on Pb results), suggesting that global contour grouping can improve on pure local methods.

3.7 Closing remarks

Understanding the mechanism for contour-completion holds the promise to develop better-performing image understanding systems. This is however, a very challenging task which is thought to be achieved by segmentation. In this chapter, we have presented a purely contour-based CRF method attempting to enforce the contour closure principle, and derived an efficient optimization method to perform approximate inference on the higher-order CRF. The results on natural scene image datasets show that our method can produce highly-completed contours without sacrificing point-wise detection accuracy. We hope this work will provide useful ideas for perceptual grouping research.

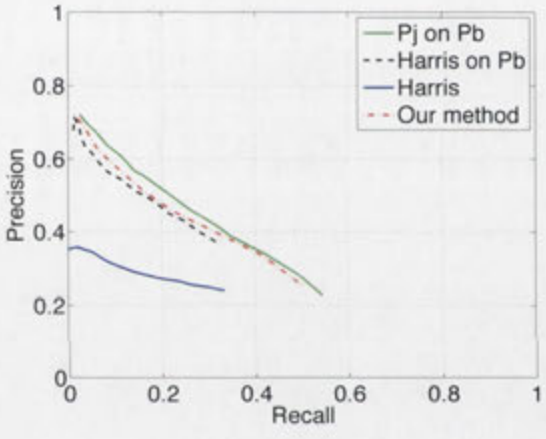


Figure 3.18: The precision-recall curves of several junction detection algorithms.

Chapter 4

Winding Number Constrained Contour Detection

4.1 Introduction

Object contours play an important role in image understanding. They not only localize the support regions of objects, but also reveal their shape which is a strong cue for recognition. Therefore, extracting a few clean and semantically meaningful contours may simplify subsequent high-level image understanding tasks [Walther et al., 2011]. However, salient contour extraction is a challenging task. First of all, the problem is highly under-constrained. Without any knowledge of objects, there are few prior constraints on the number of contours and their shape. Secondly, various factors such as texture, shadow, and lack of contrast could undermine the accuracy of contour detection.

To deal with these difficulties, a robust strategy is to use multiple types of cues. Many contour detection methods have been developed based on different cues, e.g. figure/background contrast [Shi and Malik, 2000a; Wang and Siskind, 2003; Boykov et al., 2001], contour smoothness [Schoenemann et al., 2011], closure [Andres et al., 2011; Ming et al., 2012] and symmetry [Stahl and Wang, 2008]. These cues often call for different representations. Region cues, such as the figure/background contrast, are more conveniently represented in 2D region segmentation methods [Shi and Malik, 2000a; Wang and Siskind, 2003; Boykov et al., 2001], whereas contour cues such as smoothness are better represented in edge domain. For example, methods such as the active contour method [Kass et al., 1987] focus on 1D contour detection.

Over the years, there has been a trend of jointly using contour cues and region cues for object contour detection. They can be roughly categorized as continuous or discrete methods. In continuous domain, the active contour model was adapted to use both region and contour cues [Sumengen and Manjunath] [Xie and Mirmehdi, 2004]. In discrete domain, contour cues (such as curvature) have been introduced to region segmentation methods (e.g. [Stahl and

Wang, 2007], the intervening contour approach in [Leung and Malik, 1998]). However, their techniques for cue integration are often tailored to their specific problems e.g. relying on heuristics or an unduly complicated model [Zhang and Ji, 2010], and lack explicit interaction between region and contour labels.

This chapter attempts a more general, yet efficient approach that tightly integrates both region cues and contour cues. We consider the contour extraction problem within an energy minimization framework. Our objective function is designed to encode various region and contour cues by explicitly introducing both 2D region labels and 1D contour labels. Then we use novel constraints to ensure the consistency of region and contour labels. Compared to previous methods, this framework allows for higher flexibility in choosing energy functions.

The key to this framework is the design of constraints with the following three properties. First of all, these constraints must ensure the topological correctness of solutions. For example, regions with different labels should be separated by contours, and object boundaries should not be fragmented. Second, these constraints should not be too restrictive such that there is no feasible solution. Last but not least, constraints should be encoded efficiently, and thus induce minimal computational costs, e.g., preferably a small number of linear constraints. However, it is recognized that, constraints satisfying these conditions are not easy to design. Recent work [Andres et al., 2011] argued that in order to ensure the closedness condition, exponentially many constraints are needed.

In this work, we propose a novel and simple method to describe the region-contour consistency relationship, based on the classic "winding number" concept from the field of algebraic topology [Meister, 1769]. By definition, a winding number, which involves several closed (but not necessarily simple) planar curves and a point in the plane, refers to the number of times the curves revolve around this point. Our key technique is to restrict region labels to be the winding numbers of contours, so that a small number of linear constraints can provably ensure region/contour consistency.

Our approach can be considered as an example of the "duality" relationship between regions and contours. Being dual, properties of one can be converted into properties of the other. A well-established example is the application of Green's theorem in a plane. Draw a simple Jordan curve (i.e. closed and non-self-intersecting contour) in the plane. Many quantities (such as its areas) defined on the 2D region can be computed efficiently via 1D line integral along the contour.

We apply our framework to the scenario in which regions have binary labels, and focus on the foreground contour detection. Using the winding number technique, we integrate region segmentation cues into the ratio-based contour detection formulation [Wang et al., 2005; Stahl

and Wang, 2007]. Our objective function includes three components for contour saliency, region similarity and contour smoothness, respectively, and can be efficiently solved by linear programming.

Our method is evaluated on several public datasets, and is compared with pure contour or region based approaches. Our method is also extended to incorporate user interaction for interactive image segmentation. Although this chapter focuses on middle-level perceptual grouping, we believe that our method is applicable to high-level tasks where both contour cues and region cues are helpful (e.g. object detection).

4.1.1 Problem setup and chapter overview

The input of our method is a natural scene image. The output is a closed contour which may correspond to the outline of an object. Unlike most of existing work, our method is able to integrate contour cues and region cues through the device of winding numbers. The definition of this concept is in Section 4.2. Our method is based on a framework of energy minimization, as detailed in Section 4.3. In Section 4.4, this framework is instantiated by ratio-based energy functions. Experimental results by these energy functions are shown in Section 5.6, and conclusion is in Section 4.6.

4.2 Winding number and its fast computation

In this work, contours are considered as curves in a two dimensional Euclidean plane E^2. A curve is defined as the image of a continuous map from an interval of a real line to the plane. A curve is closed if its end point coincides with the start point. A closed curve is also called a loop. The winding number of a loop about a point is defined as the number of times the loop travels around the point in counter-clockwise direction. Here, the loop does need to be *simple*, i.e. it is allowed to intersect with itself. As shown in [Gallier and Xu, 2013], any connected component of the plane has a constant and integral winding number. Therefore, we can assign a single winding number to a connected region. We further extend the winding number concept to a plane with a finite number of curves. The winding number of a region is defined as the sum of the winding numbers of all loops. Figure 4.1 shows the winding numbers of different regions induced by two closed curves. Take the region labeled with a winding number 2 for example, both two curves travel around this region in counter-clockwise direction, and each curve induces a winding number 1 to the region.

If winding numbers were to be computed according to the definition, we have to perform integration along loops in the plane. However, there is a fast method to compute winding

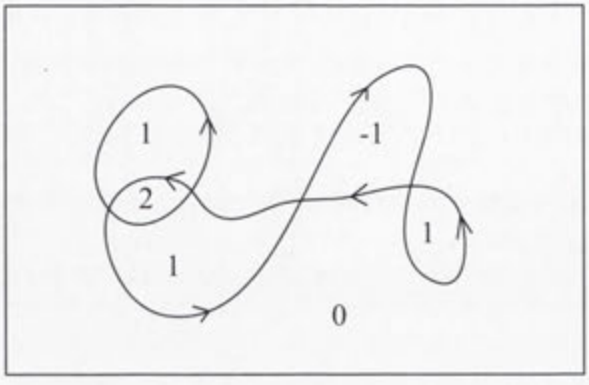


Figure 4.1: Winding numbers induced by closed contours.

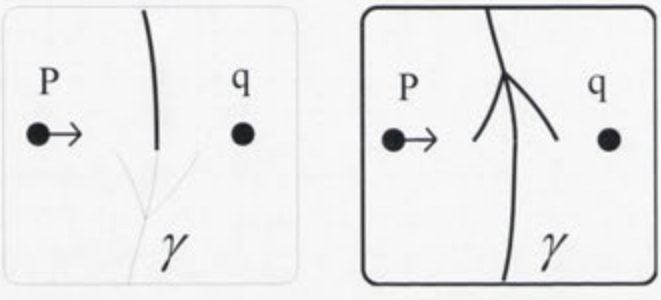


Figure 4.2: Left: $wn_p = wn_q - 1$. Right: $wn_p = wn_q + 1$. wn_p and wn_q are the winding numbers of point p and q respectively.

numbers using the *crossing rule* [Needham, 1999]: "If a loop is moving from our left to our right [our right to our left] as we cross it, its winding number around us increases [decreases] by one."

Let wn_p denote the winding number of point p, and y_α denote the binary label of edge α indicating whether the edge is active. The edge extraction step is explained in Section 4.3.1. An illustration of the crossing rule is shown in Figure 4.2.

According to the crossing rule, if we draw an path from point p to point q, the winding number difference is

$$wn_p - wn_q = \sum_{\alpha \in L_{pq}} y_\alpha - \sum_{\beta \in R_{pq}} y_\beta \tag{4.1}$$

where L_{pq} and R_{pq} denote the indices of edges crossing from right to left, and edges crossing from left to right on that path, respectively. Please note that winding numbers are solely determined by the loops, it follows that Eq (4.1) holds regardless of the choice of paths. The intuition behind can be compared to climbing a hill. Whichever path we may choose, the total number of steps upward minus the total number of steps downward should be determined by

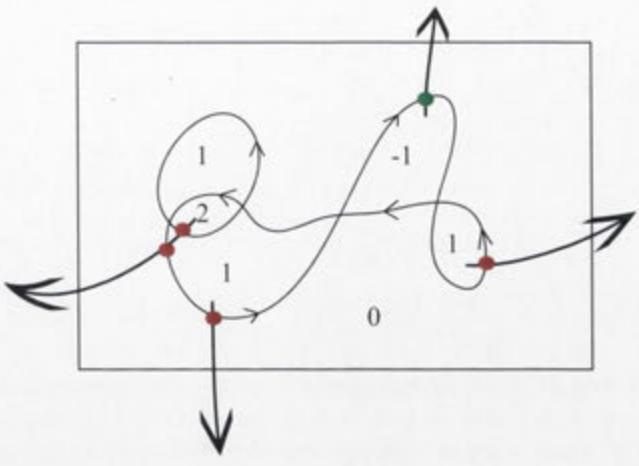


Figure 4.3: Fast winding number computation. Draw an arbitrary path to outside of the image frame, the winding number of a point equals the number of edges crossing from right (red dot) minus the number of edges crossing from the left (green dot).

the height of the hill. This property will make winding number calculation robust in experiments.

If we predetermine the winding number of some points in the plane to zero, e.g. the points very far away from any curve. Then we draw an arbitrary path starting from the inside of the region p to that point, then we can apply Eq (4.1) to calculate the winding number of the region as follows:

$$wn_p = \sum_{\alpha \in L_p} y_\alpha - \sum_{\beta \in R_p} y_\beta \tag{4.2}$$

where L_p and R_p are the edges crossing from right to left, and edges crossing from left to right, respectively. This method is illustrated in Figure 4.3.

This fast computation method (4.2) will be used to connect region labels to edge labels in our energy minimization framework.

4.3 Region-boundary consistent contour extraction

This section presents our method for region-boundary consistent contour extraction. Section 4.3.1 discusses how to extract basic edge and region hypotheses from an image. Section 4.3.2 discusses our general energy minimization framework and its consistency condition. Section 4.3.3 shows how to use winding numbers to enforce this condition in a simplified way.

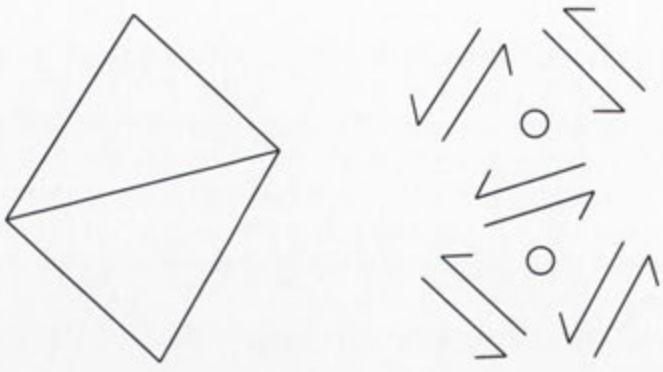


Figure 4.4: Examples of region and edge hypotheses. The left shows two triangular image regions and their edges. The right shows the edge and region hypotheses extracted from the image. Two circles denote the variables of two regions, and each arrow represents a variable of a directed edge.

4.3.1 Basic edge and region hypotheses

Salient contours are those human perceive from noisy background. Please note that they are not necessarily (part of) salient objects which would draw immediate human attention. We formulate the salient contour extraction problem as an energy minimization problem defined on both region and edge hypotheses. We choose superpixel over-segmentation [Levinshtein et al., 2009b] as a means to provide sufficient edge and region hypotheses. Each superpixel provides an atom region hypothesis. The boundary of each superpixel are linearized into a number of edge-elements. For each element, two oppositely directed (bi-directional) edge hypotheses called conjugate edges are introduced. It is important to note that our winding number formulation is not restricted to the superpixel setup, but applies to general boundary-region graphs as well. An illustration of this setup is shown in Figure 4.4. These two triangles give rise to two atom regions and eight directed edges.

Let the variable $\mathbf{x} = \{x_i | i = 1...N_r\}$ denote the labels of N_r atom regions, and $\mathbf{y} = \{y_j | j = 1...N_e\}$ denote the binary labels of N_e edges. The edge label space is denoted as $\mathcal{Y} = \{0,1\}^{N_e}$. The label space of all region variables is denoted as $\mathcal{X}$. Although the winding number concept is potentially applicable for multiple-label segmentation problems, our work focuses on the binary-label figure/ground segmentation problem. In other words, we let $\mathcal{X} = \{0,1\}^{N_r}$.

4.3.2 Energy functions and the consistency condition

The problem of salient contour extraction is considered in the energy minimization framework, as many previous work did [Kass et al., 1987; Stahl and Wang, 2007; Kokkinos, 2010a; Shi and Malik, 2000a]. The energy function $E(\mathbf{x}, \mathbf{y})$ represents various cues, such as figure-

background contrast and contour smoothness, depending on applications. Our main contribution is constraints which ensure the topological correctness of solutions. Together, the basic energy-minimization problem has the following form:

$$\min_{\mathbf{x},\mathbf{y}} \quad E(\mathbf{x},\mathbf{y}) \tag{4.3}$$

$$s.t. \quad \Phi_W(\mathbf{x},\mathbf{y}) = 0 \tag{4.4}$$

$$\Phi_C(\mathbf{x},\mathbf{y}) = 0 \tag{4.5}$$

$$\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y} \tag{4.6}$$

In order to guarantee the topological validity of labels, two sets of constraints are set up. The constraint set Φ_C is the edge continuity constraints, as follows:

$$\sum_{i \in j_{\text{in}}} y_i = \sum_{i \in j_{\text{out}}} y_i, \quad \forall j \in V, \tag{4.7}$$

where j is from the vertex index set V; j_{in} and j_{out} denote edges indices heading into and moving out of the vertex j, respectively. These constraints say that the net flow at every vertex is zero. For a flow network without source and sink, all the flows can be decomposed into a set of cycles. Therefore, our method aims to extract a set of closed curves as contours.

The constraints Φ_W denote the winding number constraints which ensure the consistency of region and contour labels. The specific consistency condition used in this chapter is that:

If an edge is active, its adjacent (i.e. incident) regions must have different region labels; if two adjacent regions have different labels, one of the edge elements in-between must be active.

This condition guarantees that every edge must be part of a region boundary, and every region is enclosed by contours. Figure 4.5 shows one correct labeling and two incorrect cases that violate the condition. At first glance, this condition does not have anything to do with winding numbers. Instead, it can be formulated as follows:

$$|y_m - y_n| = \mathbf{1}\left(x_i - x_j = 0\right), \quad \forall (i,j) \in G, \tag{4.8}$$

where x_i and x_j denote the labels of two adjacent regions, and G denotes the set of indices of adjacent regions. Variables y_m and y_n denote two conjugate edges separating these two regions. The function $\mathbf{1}(\cdot)$ equals one if its argument is true, and equals zero otherwise. Although constraints (4.8) are sufficient for the consistency condition, they are expensive to implement due to their non-linearity. Even if the energy function E is convex, the whole energy minimization problem will generally turn out to be non-convex with these non-linear constraints. Next

Figure 4.5: The left image shows a consistent region and edge configuration. The middle figure is not consistent because two regions separated by a contour have the same label. The right figure is not consistent because two adjacent regions with different labels are not separated by any contour.

section will show that this condition is guaranteed by linear constraints based on the winding number concept.

4.3.3 Winding number constraints

We realize that the winding number concept, from topological study, provides an elegant and effective means to parameterize the region-contour consistency condition for image segmentation. We have reached the following key step called *winding number technique*:

The label of a region can be identified by its winding number induced by contours.

The winding numbers can be used as region labels due to properties discussed in Section 4.2. For example, region labels need to be integral and have the same value in connected regions. These conditions are proved to be true for winding numbers. More importantly, the winding number constraints guarantee the consistency between region labels and contour labels. First of all, due to Eq (4.9), the winding numbers of adjacent regions will be different if one of the conjugate edges between them is active.[1] Secondly, Eq (4.9) also suggests that two regions which are not separated by any edges must have the same winding number. In other words, these two regions cannot have different labels. Thirdly, the existence of winding numbers also guarantees the feasibility of our problem (4.3). In conclusion, this winding number scheme does encode the region-contour consistency condition.

The benefit of such winding number scheme also lies in that: it leads to a small number of linear constraints. This can be made evident by examining the fast computation procedure of winding number computation in Eq (4.2). To adapt it to our problem, we assume that every image is enclosed by a rectangular border, any region outside of the image has a label zero. Therefore, the winding number of region i is computed as:

$$x_i = \sum_{\alpha \in P_i} y_\alpha - \sum_{\beta \in N_i} y_\beta, \quad \forall i \tag{4.9}$$

[1] If both of the conjugate edges are active, the two regions must share the same label in the same way as when both edges are inactive.

Figure 4.6: The left shows an image in BSDS dataset. The right shows paths by which the winding number of the superpixels (red dots) are calculated.

where P_i and N_i are the edges crossing from right to left, and edges crossing from left to right, respectively. Eq (4.9) for all atom regions together can be represented as the following winding number constraint, denoted as Φ_W in Eq (4.4):

$$\mathbf{x} = M\mathbf{y}, \tag{4.10}$$

where M is a matrix whose entries are 0, 1, or -1. Take the i-th row of M for example, $M(i,\alpha) = 1, \forall \alpha \in P_i$; $M(i,\beta) = -1, \forall \beta \in N_i$; and the rest of entries are zero. The number of these constraints is the same as the number of atom regions.

The winding number constraints in theory do not depend on the choice of path from each atom region to the outside of the image. However, determining the adjacency relationship numerically can be difficult in the neighborhood of junctions. To minimize the risk of incorrect adjacency estimation, our method chooses a path such that the total number of regions on this path is small, and the edges crossing this path are long. Some examples of paths are shown in Figure 4.6.

Since region labels are determined by edge labels linearly, the winding number constraints may restrict the set of feasible region labels. However, if region labels are binary, the following proposition shows that the winding number constraints do not restrict solutions of segmentation at all.

Proposition 1. *For any segmentation in which region labels can only be zero or one, there always exists a set of oriented boundaries such that the regional labels equal the winding numbers induced by these boundaries.*

Proof. First of all, we assume that edges do not intersect with each other and each edge is

only adjacent to two regions. If this assumption is not valid, the edges can be divided into smaller segments to satisfy the assumption. Then, for an atom region whose label is one, we set a cycle of its adjacent edges in counterclockwise direction to be active. This cycle of edges will induce a winding number one to this region, and a winding number zero to other regions. Since edges are not shared by more than two regions, this operation can be done to every atom region without conflict. Consequently, every atom region in the foreground has a winding number one. Last, conjugate edges which are both active can be removed without affecting winding numbers of any region. Therefore, the final contour map is consistent with the given segmentation. □

4.4 Application to ratio-based energy functions

Many energy functions have been proposed for image segmentation and contour grouping. Here, we pay special attention to the ratio-based contour detection and segmentation methods which have been studied in [Wang et al., 2005; Stahl and Wang, 2007; Levinshtein et al., 2010b; Schoenemann et al., 2011]. One advantage of ratio-based energy functions is that they have little bias towards either large or small regions. Also, they are not biased towards equal partitions as normalized cuts does. Another advantage is that these energy functions are relatively easy to optimize.

Therefore, ratio-based energy functions are used as an example to demonstrate the effectiveness of our winding number technique. Section 4.4.1 shows how contour cues are integrated with region similarity cues through our constraints. Section 4.4.2 shows how the curvature cue is integrated. Section 4.4.3 extends our method to incorporate user interaction. Implementation details are presented in Section 4.4.5. We also discuss other energy functions in Section 4.4.6.

4.4.1 Incorporation of region similarity cues

The contour-based energy function our method adopts is the ratio between contour gaps and areas of foreground, defined as ([Stahl and Wang, 2007]):

$$\frac{E_B(\mathbf{y})}{A(\mathbf{x})} \tag{4.11}$$

The boundary term measures the total gap length within contours:

$$E_B(\mathbf{y}) = \alpha_b \sum_i v_i y_i \tag{4.12}$$

where v_i is the gap length in edge i. The parameter α_b controls strength of the boundary term. This term will favor foreground objects with salient boundaries. The denominator is the total areas of foreground:

$$A(\mathbf{x}) = \sum_i a_i x_i \tag{4.13}$$

where a_i is the area of region i. In [Stahl and Wang, 2007], areas are converted into second edge weights of a graph, and an optimal solution is obtained by solving a graph cycle-finding problem. A problem with Eq (4.11) is that there may be strong distracting contours inside an object or in background. Here a region similarity term E_R is added to improve robustness to such noise. The new objective function is defined as:

$$E(\mathbf{x}, \mathbf{y}) = \frac{E_R(\mathbf{x}) + E_B(\mathbf{y})}{A(\mathbf{x})} \tag{4.14}$$

The new addition, the region term $E_R(X)$ is defined as the sum of the affinity measures between figure and background superpixels:

$$E_R(\mathbf{x}) = \alpha_r \sum_{(i,j)\in P_R} w_{ij}|x_i - x_j| \tag{4.15}$$

where P_R denotes pairs of regions whose distance is smaller than a prescribed threshold. The weight w_{ij} encodes the color difference between regions i and j. α_r is a parameter to control the strength of the region term. This term favors large figure-ground contrast.

To ensure region-contour consistency, we use three sets of constraints. The first two sets of constraints are the continuity constraints Eq (4.5) and the winding number constraints Eq (4.4). They have been discussed in Section 4.3. Following the binary label assumption, we limit any region and edge label to be binary. In sum, our ratio-based segmentation model is as follows:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & \frac{E_R(\mathbf{x}) + E_B(\mathbf{y})}{A(\mathbf{x})} \\ s.t. \quad & \Phi_W(\mathbf{x}, \mathbf{y}) = 0 \\ & \Phi_C(\mathbf{y}) = 0 \\ & \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y} \end{aligned} \tag{4.16}$$

where the label spaces are defined as $\mathcal{X} = \{0,1\}^{N_r}$ and $\mathcal{Y} = \{0,1\}^{N_e}$. Although Eq (4.16) is good enough for ensuring the region-contour consistency, the formulation can be further simplified by replacing region labels with edge labels using Eq (4.10), which leads to an energy function of edge variables.

4.4.2 Incorporation of curvature cues

Recognized as the Gestalt law of good continuity, the human vision system has a preference for grouping smooth contours together. Our method can be extended to take into account of contour smoothness. The smoothness of contours can measured by integral of squared curvature. Let P_E denote the indices of all pairs of edges sharing one vertex. The binary junction variable z_{ij} is associated with the junction formed by edge y_i and y_j. Let $\mathbf{z} = \{z_{ij}|(i,j) \in P_E\}$ denote all N_j junction variables, and $\mathcal{Z} = \{0,1\}^{N_j}$ is the label space of junction variables. In our model, the total curvature cost is defined as:

$$E_C(\mathbf{z}) = \alpha_c \sum_{(i,j)\in P_E} u_{ij} z_{ij} \tag{4.17}$$

where the parameter α_c controls the strength of the curvature term as a whole. The curvature weight u_{ij} is the sum of squared curvature along both edges. In [Levinshtein et al., 2010b], only curvature costs within edge fragments are taken into account. However, our curvature term also penalizes sharp turns at junctions.

Our junction model is illustrated in Figure 4.7. To ensure correct junction configurations, junction constraints are devised. They are denoted as $\Phi_J(\mathbf{y},\mathbf{z}) \leq 0$. These constraints are adapted from the connectedness constraints for undirected edges [Ming et al., 2012]. These constrains consist of two parts. First, every active edge should form transition to at least one edge whose tail connects to the head of the current edge. Second, every junction variable can be active only when both of its associated edges are active. The junction constraints are translated into the following linear inequities:

$$\sum_{j|(i,j)\in P_E} z_{ij} \geq y_i, \quad \forall i \tag{4.18}$$

$$z_{ij} \leq y_i, \quad \forall (i,j) \in P_E \tag{4.19}$$

$$z_{ij} \leq y_j, \quad \forall (i,j) \in P_E \tag{4.20}$$

The inequities (4.18) correspond the first part of junction constraints. The inequities (4.19) (4.20) correspond to the second part of junction constraints. In sum, the energy minimization problem

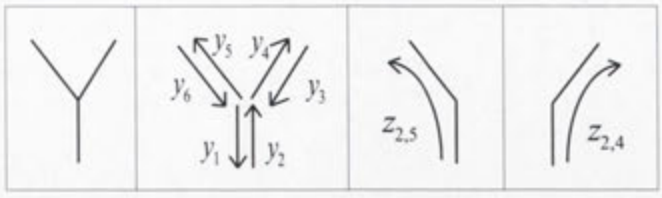


Figure 4.7: Our junction model. The first figure shows one junction detected in the image. The second figure shows the 6 variables representing the associated edges. The third and forth figures show two possible L-junctions if edge y_2 is active.

Figure 4.8: An example in which the curvature term affects our model's output. The first image is an input image. The second image shows our method's output without the curvature term. Last image shows the output under the influence of the curvature term. (Best viewed in color.)

is as follows:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y},\mathbf{z}} \quad & \frac{E_R(\mathbf{x}) + E_B(\mathbf{y}) + E_C(\mathbf{z})}{A(\mathbf{x})} \\ s.t. \quad & \Phi_W(\mathbf{x},\mathbf{y}) = 0 \\ & \Phi_C(\mathbf{y}) = 0 \\ & \Phi_J(\mathbf{y},\mathbf{z}) \leq 0 \\ & \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}, \mathbf{z} \in \mathcal{Z} \end{aligned} \tag{4.21}$$

The effectiveness of adding curvature terms is illustrated by the example in Figure 4.8. The left is an input image. A star shape is favored by the boundary term and the region term due to its stronger contrast to background. Therefore, our model without the curvature term chooses the star as shown in the middle of Figure 4.8. When the curvature term is added, the smoother disk is extracted as shown in the right of Figure 4.8.

4.4.3 Incorporation of user interaction

Our method described so far only uses low and mid-level grouping cues. It is recognized that the use of high-level information can make a grouping process more robust [Arbelaez et al., 2011]. A major source of high-level information is user interaction. Our method can be conveniently extended for this purpose. We assume that an user has specified some of edge and region labels by clicking on an input image. Our model can represent these inputs as

constraints. For examples, let R_F stand for the regions which the user considers as foreground, and R_B denote those considered as background. The corresponding constraints are:

$$x_i = 1, \quad \forall i \in R_F \tag{4.22}$$

$$x_i = 0, \quad \forall i \in R_B \tag{4.23}$$

4.4.4 Inference by linear relaxation

The energy minimization problems Eq (4.16) and Eq (4.21) are nonlinear integer programs. In the following, they are relaxed into linear programs which can be solved in polynomial time. Since Eq (4.16) is a special case of Eq (4.21), we only need to discuss the latter. First, the domain of all labels is relaxed to be interval $[0,1]$. Second, note that all terms in the objective function are linear except for E_R which is the sum of absolute values, according to Eq (4.15). Each absolute value $|x_i - x_j|$ is replaced by a variable t_{ij}, and two constraints which are $t_{ij} > x_i - x_j$, and $t_{ij} > x_j - x_i$. Then our model becomes a standard linear fractional program (4.24). Let the real-valued vector ξ here stand for all the variables, i.e. $\xi^T = [\mathbf{x}^T\ \mathbf{y}^T\ \mathbf{z}^T\ \mathbf{t}^T]$.

Since the denominator representing total areas is strictly positive, the fractional program can be transformed into a linear program [Boyd and Vandenberghe, 2004]. This method is also used in contour grouping work [Kokkinos, 2010b]. In general, the linear fractional program is written as follows:

$$\begin{aligned} \min_{\xi} \ & \frac{c^T\xi + d}{e^T\xi + f} \\ & A\xi = b \\ & \xi \geq 0 \end{aligned} \tag{4.24}$$

where A to f are constants. The denominator is positive, i.e. $e^T\xi + f > 0$. Let $\eta = \frac{\xi}{e^T\xi+f}, \tau = \frac{1}{e^T\xi+f}$, then the equivalent linear program is:

$$\begin{aligned} \min_{\eta,\tau} \ & c^T\eta + d\tau \\ & A\eta = b\tau \\ & \eta \geq 0 \\ & e^T\eta + f\tau = 1 \end{aligned} \tag{4.25}$$

The solution of the fractional program can be obtained as $\xi = \eta/\tau$. In general, ξ is not necessarily integral. However, in our experiments, solutions are usually very close to be integral.

4.4.5 Implementation details

The boundary gap measure v_i in Eq (4.12) equals the number of edge pixels in the segment minus the sum of the probabilities of each edge pixel being a true contour point. The probability is estimated according to [Levinshtein et al., 2010b]. The region affinity measure w_{ij} in Eq (4.15) is the sum of affinity values of all pairs of pixels in these two regions, i.e. $w_{ij} = \sum_{pq|p\in i,q\in j} w(p,q)$. The pixel-wise weight $w(p,q)$ is computed based on the similarity of colors and locations using a RBF kernel.

We use LP_SOLVE library to solve the linear programming problem. An image is usually oversegmented into about 300 superpixels, and our algorithm consists of ten to twenty thousand variables and several thousands of constraints. LP_SOLVE solves the problem in about twenty seconds on a modest laptop with Intel 2G Centrino 2 core processor/3G RAM.

4.4.6 Extension to other objective functions

Other well established objective functions such as normalized cuts can also be transformed into energy functions based on edge labels. Although this objective function is more difficult to optimize, it is included in this chapter for discussions. The objective function of normalized cuts [Shi and Malik, 2000a] is defined as:

$$\min_{\mathbf{x}\in\mathcal{X}} \frac{\sum_{ij} w_{ij}|x_i - x_j|}{(\sum_i w_i x_i)(\sum_i w_i(1-x_i))} \tag{4.26}$$

where w_{ij} is affinity between superpixel i and j. The parameter $w_i = \sum_j w_{ij}$ denotes the volume of x_i. $\mathbf{x}$ denotes all the region labels. Our transformation leads to the following edge-based segmentation problem:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \frac{\sum_{ij} w_{ij}|m_i^T\mathbf{y} - m_j^T\mathbf{y}|}{(\sum_i w_i m_i^T\mathbf{y})(\sum_i w_i(1-m_i^T\mathbf{y}))} \\ s.t. \quad & \Phi_C(\mathbf{y}) = 0 \\ & M\mathbf{y} \in \mathcal{X} \\ & \mathbf{y} \in \mathcal{Y} \end{aligned} \tag{4.27}$$

4.5 Experiments

Our method is compared with several related methods on the Weizmann horse dataset [Borenstein and Ullman, 2002a] and the Weizmann segmentation dataset [Alpert et al., 2007]. Then

Figure 4.9: Sample images from the Weizmann horse dataset. The horse images are in the first and third columns, and the corresponding groundtruth contours are in the second and forth columns.

our method is extended to incorporate user interaction which is effective for the more complex BSDS dataset [Arbelaez et al., 2011].

4.5.1 Tests on the Weizmann horse dataset

As shown in Figure 4.9, this image set contains salient horses in the middle of images. However, obtaining complete contours of horses is still challenging due to several reasons. For example, there are strong distracting contours in background and inside horse regions. True contours on the other hand may be faint or missing because of low contrast. Since horses usually are not camouflaged, we expect the incorporation of region cues to be helpful in obtaining cleaner contours.

Our method is compared with several closely related methods. To demonstrate the effectiveness of cue combination, the region-based normalized cuts method (Ncuts) [Shi and Malik, 2000a] and the superpixel grouping method (SC) [Levinshtein et al., 2010b] are used for qualitative comparisons. Ncuts is based on pairwise regional affinities, similar to the region term used in our method. SC implements the ratio contour method [Stahl and Wang, 2007] (RC) in superpixel domain, and our method uses the same superpixel implementation as SC. For quantitative comparisons, we also include RC, and GPAC [Sumengen and Manjunath] which is one of the best variational methods for cue combination.

Both SC and our method are initialized on Pb detection results. Ncuts and GPAC implementations are obtained from authors' websites. The parameters of our method are adjusted on a validation set. The results are shown in Figure 4.10. SC outputs ten solutions for each image, the one with the highest F-value is shown in the figure. We can see that our outputs better separate the horse regions from background. In SC's outputs, horse legs are often connected as a single blob region. However, region similarity cues used in our model help distinguish background regions from foreground regions. Two-way Ncuts often cuts out a homogeneous background area. Ten-way Ncuts, however, tends to produce spurious edges (e.g. those in the sky and grass). These results show that either contour cues or region cues alone are not enough for segmenting salient foreground regions. Note that our results in Figure 4.10 appear to be a single contour due to our choice of the objective function. The winding number constraints,

however, do not require the solution to be a Jordan curve.

We quantitatively evaluate related methods using F-measure on 100 horse images. The F-value of each solution is computed by comparing its segmentation mask with the groundtruth mask. The F-value is close to 100% if and only if the detected contours are close to the groundtruth contours, and the F-value declines gradually when the output contours deviate from groundtruth. For methods producing more than one solutions per image (e.g. SC, Ncuts), the overall F-value of an image is the best F-value of all the solutions. Figure 4.11 shows the F-values of SC reported in [Levinshtein et al., 2010b] which converges to 79.7% with 10 solutions per image. In our experiment, it converges to 76.5%. However, in either [Levinshtein et al., 2010b] or our experiment, the performance is much worse when the number of solution is small. Our model outputs only one solution for each image and achieves an F-value of 74.1%. RC reaches an F-value of 68.1%. We let Ncuts to produce two or ten solutions for one image by partition an image into two or ten segments. The F-value of two-way Ncuts is 43.7% while that of ten-way Ncuts is 64.9%. The F-value of GPAC is 54.5%. Using the boundary term alone, our method achieves an F-value of 64.8%, demonstrating that the region homogeneity cue is effective for this dataset. Note that with additional constraints [Carreira and Sminchisescu, 2012], our method could output more solutions at the expense of increased computation time.

4.5.2 Tests on the Weizmann segmentation dataset

Our method is also tested on the Weizmann segmentation dataset [Alpert et al., 2007] which contains one hundred single-object images and one hundred images with two objects. Our method is adjusted on the two-object images and is tested on the single-object images. We follow the same protocol as in the previous section. Qualitatively, our method can produce good results as shown in Figure 4.12. SC, Ncuts, RC, and GPAC methods are used in the comparison by F-measure. When compared with the best of ten solutions, our method's single solution F-value (78.02%) is inferior to SC (87.19%), and is comparable to RC (77.82%). However, as shown in Figure 4.13, our method has clear advantage for the single output. In addition, for some images our method can produce better outputs than the best solutions of SC and Ncuts. The F-value of two-way Ncuts is 45.7%, and that of ten-way Ncuts is 56.8%. The F-value of GPAC is 63.6%. We have analyzed the contributions of each term. When there is only the boundary term in the objective function, the F-value of our solutions reduce to 76%. Our method using the boundary term and the region term but not the curvature term achieves 78%. The improvement by the curvature term is largely in details which do not affect F-value much.

Figure 4.10: Comparisons with the superpixel closure method (SC) and the normalized cuts method (Ncuts). The first column shows the input images. The second column shows our results. The third column shows the SC results. Only the best solution (with the highest F-value) of each image is shown out of 10 solutions. The fourth column shows the 2-way segmentation results by Ncuts. The fifth column shows 10-way segmentation results by Ncuts. The last column shows GPAC results. (Best viewed in color.)

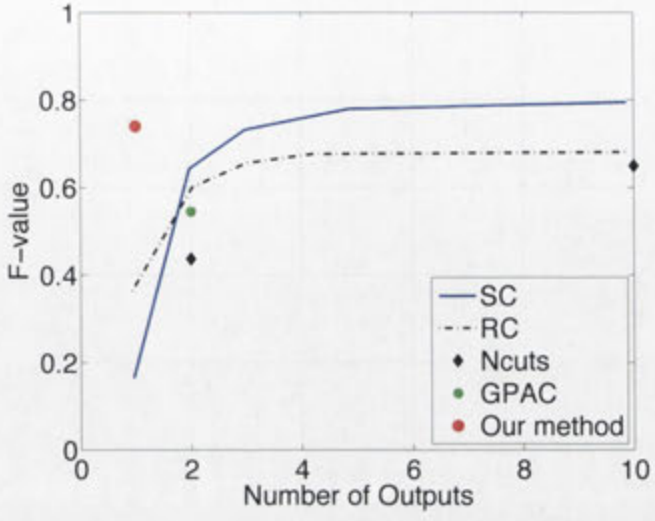


Figure 4.11: The F-values of related methods on the Weizmann horse dataset. SC method achieves highest F-value with 10 solutions. However, our method is better when considering only one solution. (Best viewed in color.)

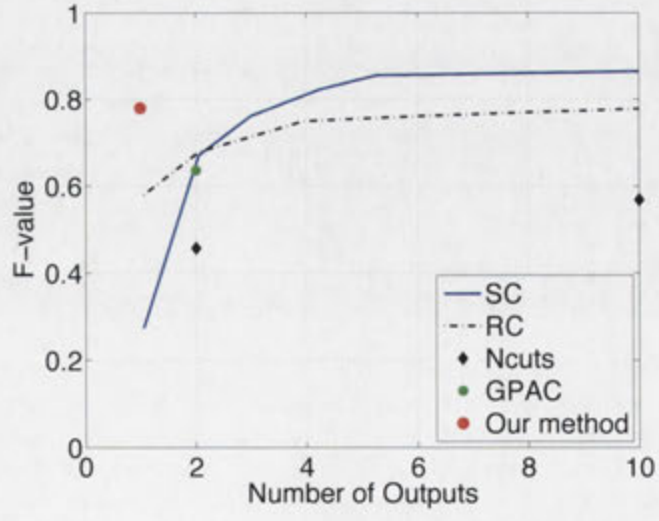


Figure 4.12: The F-values of related methods on Weizmann segmentation dataset (single-object images).

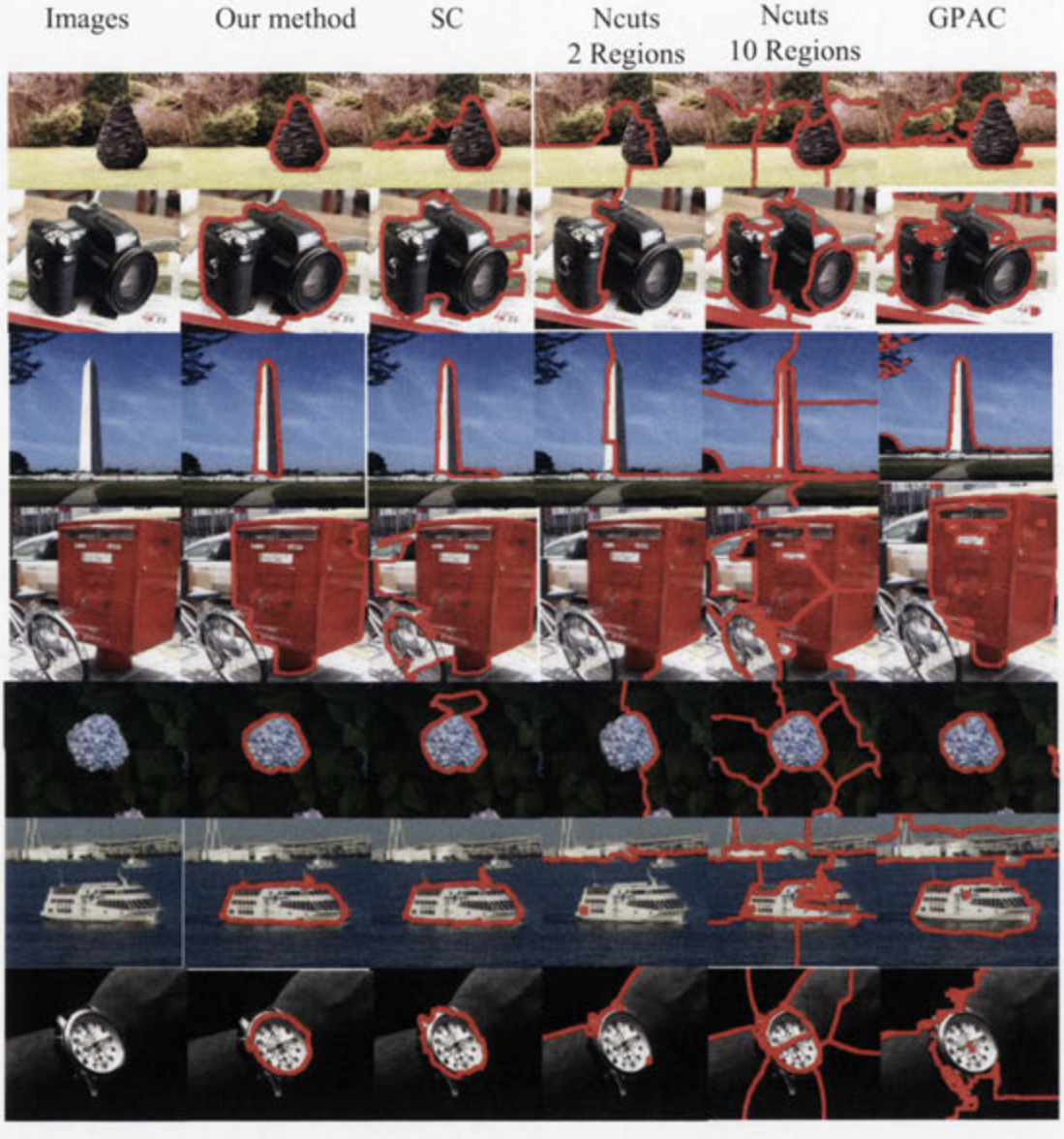


Figure 4.13: Some images in Weizmann segmentation dataset for which our method's contour outputs are either more concentrated on the objects (row 1 to row 5) or smoother (last two rows), due to the influence of the region and curvature terms. The first column shows the input images. The second column shows our results. The SC results displayed in the second column are the best ones (in terms of F-value) out of ten solutions. The Ncuts results are shown in fourth and fifth columns. GPAC results are in the last column.

4.5.3 Incorporating user interaction on the BSDS300 dataset

To demonstrate the effectiveness of incorporating user interaction, our baseline method is firstly evaluated on the BSDS300 dataset which contains a variety of urban and natural scene images. The BSDS dataset is more challenging because many images contain multiple foreground objects and background clutter. Some sample results are shown in Figure 4.14, and they match our perception of salient contours. As shown in the first row of Figure 4.15, the addition of region cues sometimes bias results towards homogenous parts of objects. When several objects have similar colors, our solution often cannot differentiate these objects, as shown in the second row. In addition, we observe the difficulty of detecting camouflaged objects for which neither cue is effective. The satisfactory results are only produced for images in which the foreground object/region is evident. Overall, these observations confirm that our results indeed reflect both region and contour cues but high-level cues are needed for more accurate results.

Our method is extended to use interaction as follows. The automatic contour detection result of an input image is first shown to a user. An interface is then provided to record the user's clicks on the input image. Mouse positions of left clicks are assumed to be in the figural region, and those of right clicks to be in the background. We detect the atom regions which are clicked on. The labels of these regions are constrained to be zero or one depending on the type of clicks. These constraints derived from user interaction are added into the system and a new solution is obtained and presented to the user. The user can provide additional inputs until the solution becomes satisfactory. Figure 4.16 shows that our method can find solutions which are consistent with human inputs. Clear improvements are observed on these challenging images due to the high-level inputs.

Results by popular interactive segmentation methods Grabcuts [Rother et al., 2004] and Random Walker [Grady, 2006] are also shown for comparison, using codes implemented by Blumenthal[2], and [Andrews et al., 2010], respectively. The quality of a segmentation result is measured by an index which is the intersection of the segmentation mask and its groundtruth mask over the union of two masks. The value of this index is between zero and one. For these four images, the average indices of our method, Grabcuts, and Random Walker are 0.89, 0.87 and 0.76, respectively. Grabcuts has comparable performance to our method. However, labeling a bounding polygon is more laborious than labeling several seeds. Random Walker also uses seeds as input. However, it requires a larger set of seeds in order to obtain good results for those images. In this experiment, the seed set for Random Walker is a superset of ours. Besides human input, our method can be extended to use object detector's output.

[2] http://grabcut.weebly.com/index.html

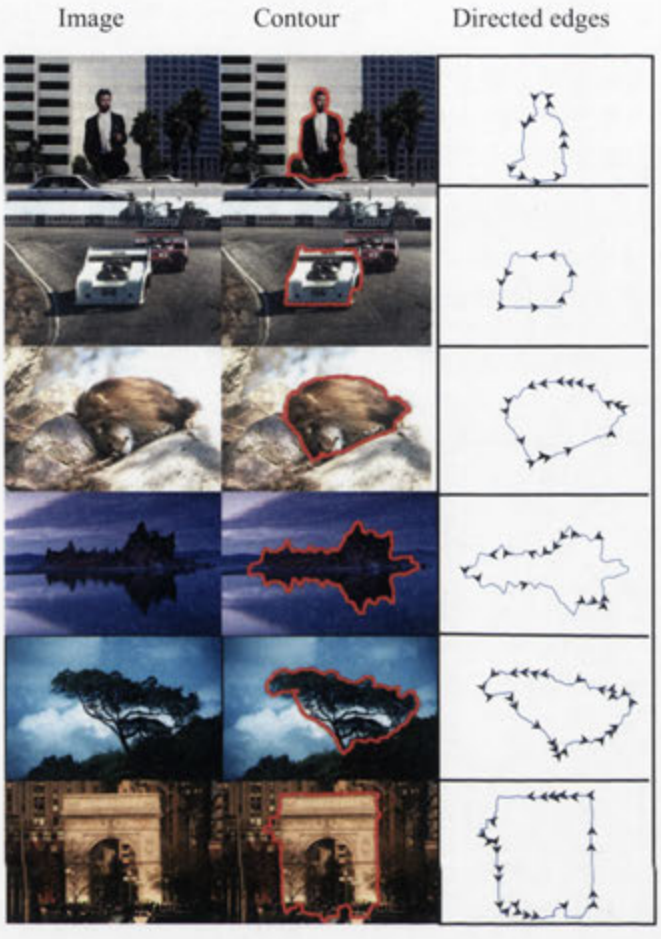


Figure 4.14: Sample results on BSDS300 dataset. The first column is the input images. The second column is the output contour overlaid on the input images. The third column shows the directed active edges. (Best viewed in color.)

Figure 4.15: Examples of segmentation bias resulted from the homogeneity cue. For images in the first row, the extracted contour focused on homogenous regions rather than whole objects. For images in the second row, our method cannot separate different objects with similar color. (Best viewed in color.)

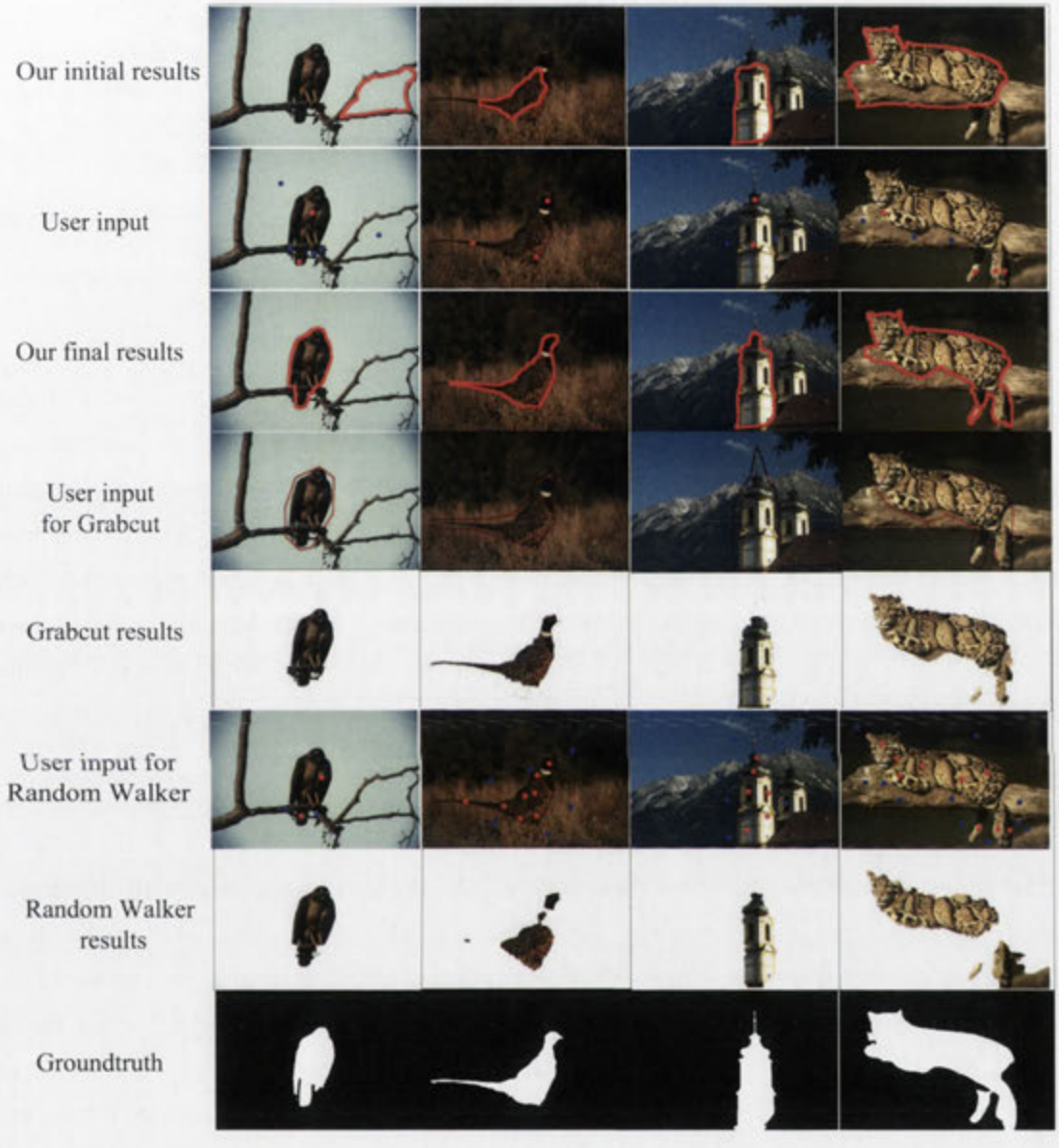


Figure 4.16: Improved contour extraction results by user interaction. The first row shows our outputs not using interaction. The second row shows user inputs. The red dots denote the regions in foreground and the blue dots denote the regions in the background. The third row shows our outputs under the guidance of the user inputs. The user input for GrabCut, and its results are shown in the forth and fifth rows, respectively. The next two rows show input for Random Walker and its results. The last row shows the groundtruth segmentation masks. (Best viewed in color.)

Figure 4.17: Different user inputs (first row), and the corresponding segmentation results (second row) by our method.

4.5.3.1 Seeds selection

Our method's results depend on the user input, as shown in Figure 4.17. However, all these results show improvements over the one without interaction. This demonstrates the robustness of our method under input variations. An important question is how to select an optimal set of seeds. Our experience is that the seeds are better placed near the boundaries where segmentation is not accurate. To save human effort, our interactive segmentation method can feedback the current segmentation result to the user after every click. In this way, a user can correct segmentation errors greedily. In contrast, a user must provide all polygon vertices to Grabcuts before getting any feedback.

4.6 Conclusion and future work

This chapter introduces a method for combining region and contour representations efficiently, based on the winding number concept. This model is simple and appealing, as it only involves a compact set of linear constraints to ensure the consistency of both representations. As an application of this method, region similarity cues and region-based user interaction are added into our ratio-based contour detection framework, and lead to improved results. This method is further extended to incorporate user interaction as a special kind of region cues. In future, we believe that more sophisticated design of region/contour cues could help to extract contours of complicated objects. We are also interested in finding efficient optimization methods for other objective functions and extend this method to the multiple-label case.

Chapter 5

A GPLVM Framework for Top-down Guided Category-level Edge Detection

5.1 Introduction

Human eyes can easily extract edges that belong to certain classes of objects from a cluttered background. This is because objects from the same class share some common and characteristic edge patterns which distinguish them from other object classes. The chapter aims to develop a computational model to address the following question: given category-level top-down guidance information, how to solve the edge detection problem more accurately.

Specifically, we plan to design a "smart" edge detector that is able to detect edge maps corresponding to specific types of objects. Our new edge detector is able to extract category-specific edge maps robustly with respect to background clutters and noise, as well as to intra-class shape variations. Finding meaningful edges in an image is of fundamental importance for performing high-level vision understanding tasks. An edge map furnished with category-level semantic meanings not only facilitates object segmentation, but also helps extract important object attributes such as pose or orientation. Figure-1 shows using category-level top-down guidance our new edge detector produces a much cleaner edge map that highlights the horse category.

Our framework is based on supervised learning, and is able to handle two different scenarios or settings i.e., supervised and weakly-supervised. In the supervised setting, we assume that manually labeled edge maps of certain categories are available for training. In the weakly supervised setting, we assume no such labeled edge maps are available, therefore our method must learn from raw input images. In this case, all the training information we have is that input images are sampled from the same category. This task is similar to the Active Basis Model (ABM) in [Wu et al., 2010c]. Our framework has been applied to both settings, and obtains promising results.

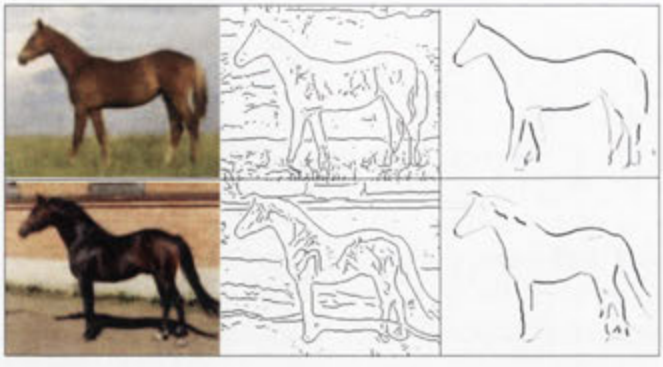


Figure 5.1: From left to right: input images, thresholded gPb edge maps, and our results.

For category-level edge detection, the main difficulty is how to build an efficient edge-map representation that is flexible enough to accommodate both (1) global shape variations, and (2) local deformation, as well as (3) allows for efficient combination of top-down and bottom-up information. We now elaborate on these three issues in sequel.

The first issue is about global shape variations. For example, consider the edge maps of human faces, which vary notably due to differences in poses, expressions, identities. Previous methods to solve this issue are either based on inflexible shape assumptions e.g. [Wu et al., 2010c], or rely on sophisticated mechanisms such as that presented in e.g. Shape Boltzmann Machine [Eslami et al., 2012], or Inverse Edge Detector [Hariharan et al., 2011]. Although these methods demonstrated improved shape modeling capacities, their training processes and inference algorithms are typically very involving, preventing them from wide applications. For instance, training the inverse detector for the pedestrian class requires training hundreds of human part detectors. In addition to expensive training, all these methods often demand a large amount of training samples which are expensive to collect. Our work strikes to achieve a balanced framework which is easy to train, able to learn large shape variations, and less demanding of training samples.

A key innovation of this work is the assumption that shape variations of object edge maps within a certain category can be modelled by Gaussian processes. We therefore adopt the Gaussian Process Latent Variable Model (GPLVM) [Lawrence, 2005] to represent the stochastic distribution of category-level edge maps. Our method is inspired by [Prisacariu and Reid, 2011b] which demonstrates that the non-linear manifold space of shapes (as encoded by binary shape masks) can be effectively modeled by Gaussian processes. Moreover, the learned shape model can then be used to aid tasks such as object detection and object tracking. This success has motivated our extension to the edge-map extraction task. In doing this, there are however nontrivial challenges, one of which is the above mentioned edge representation prob-

lem. The other is how to map back from the learned model to a test image. Thanks to the non-parametric property of GPLVM, its learning capacity grows linearly with the amount of training data. Moreover, it provides explicit mapping from a latent space to a feature space, which is very convenient for top-down inference. In our context this means, mapping from a high level latent space to a low-level edge map on the test image is practically doable. To the best of our knowledge, this chapter is the first which applies GPLVM for the edge detection task.

The second issue concerns with local shape deformation. GPLVM cannot be directly applied to binary edge maps, because edges are often distributed sparsely in the images, hence a small deformation will result in large distance between two edge maps (as noticed in [Prisacariu and Reid, 2011b]). To overcome this, [Prisacariu and Reid, 2011b] adopted the Fourier descriptor to parameterize shape masks. However Fourier descriptor is only applicable to a shape mask formed by a topologically simple curve (i.e. a singly closed contour). In our case, an edge map can have more complex topologies. Another method [Cremers, 2006] employed signed distance maps to handle this problem. However, the non-linear distance transform is sensitive to noisy edges. In the rest of the chapter, we will explain our novel solution – which is based on linear pooling variables as will be explain later in more detail. Our solution is more robust to local deformation, and is applicable to edge maps with an arbitrary topology.

The third issue is how to efficiently combine top-down and bottom-up information for optimal prediction. We firstly take advantage of the linear relationship between pooling variables and edge labels to simplify our energy function. In addition, an alternative optimization procedure is developed to decompose our high dimensional non-convex inference problem into a sequence of low-dimensional non-convex problems and high-dimensional convex problems.

All these innovations are integrated into a three-layer CRF framework, which will be detailed in Section 5.2. In experiments, we demonstrate that non-linear shape variations are effectively captured in the GPLVM latent space. Compared with methods without top-down information, our method highlights the most important category-characteristic edges while suppresses many of irrelevant foreground or background clutters. Our learned prior results in clear improvement over bottom-up edge detectors such as [Zheng et al., 2010] [Ren et al., 2005], in terms of F-value. In contrast to the level-set based methods [Prisacariu and Reid, 2011b], our method can describe both closed contours and fragmented contours. Compared with ABM [Wu et al., 2010c]–a state of the art method for weakly supervised object sketching, our method allows for higher degree of intra-class shape variations, thus outputs edge maps that are more relevant to the category of interest.

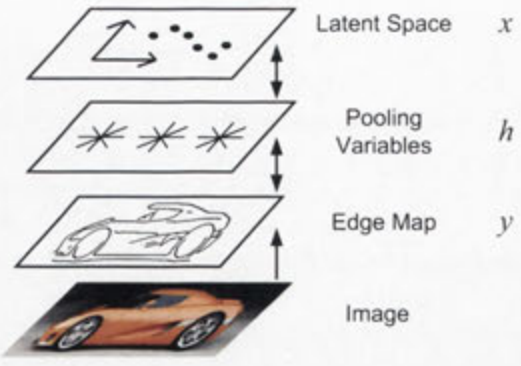


Figure 5.2: A hierarchical framework for edge map representation. Given an input image, the bottom level consists of edge hypotheses extracted from the image. The mid-level is HOG-like linear pooling variables. At top level, the edge map is encoded as the coordinates in a low dimensional latent space.

5.1.1 Problem setup and chapter overview

Our method works in both supervised scenario and weakly supervised scenario. In the first case, our method outputs a category-specific edge map for an image containing interested category. Our method is trained on clean edge images. In the second case, our method produces edge maps for a group of images. The assumption is that all these images contain objects from the same category. The foundation of our method is a hierarchical edge map representation discussed in Section 5.2. The learning and inference procedures are presented in Section 5.4 and Section 6.3, respectively. The results for supervised case are in Section 5.6. This chapter is concluded in Section 5.8.

5.2 A hierarchical edge map representation

Our method expects edge maps to be roughly aligned, e.g. with the help of object detectors. An edge map is represented at three levels as shown in Figure 5.2. At bottom level, an edge map is simply generated by gPb or Pb [Arbelaez et al., 2011] as edge hypotheses/proposals. At middle level, a novel type of linear pooling variables is used to pool information within a small neighborhood to achieve variation robustness or resilience. At top level, a latent variable model is used to encode high-level edge information in a low-dimensional latent space. Below we will briefly explain each of the three levels, and leave their detailed formulations (which is based on conditional random field-CRF) in Section 5.3.

5.2.1 Bottom-level edge hypotheses

At bottom level, an edge image is represented as edge hypotheses. In the supervised case, edge hypotheses are the line segments produced by bottom-up edge detectors [Arbelaez et al., 2011], or by human labeling. In the weakly supervised case, the edge hypotheses may be produced by uniformly sampling input images, at different locations and orientations. Our goal is to assign a binary label to each edge hypothesis, to indicate whether or not it should lie on the final edge map.

5.2.2 Mid-level linear representation

Since a bottom-level edge map is very sensitive to local shape deformation, it is difficult to directly compare two binary edge maps. For example, if we treat each edge map a long (binary) vector, and attempt to compare them directly via vector inner product, then the result is very likely to be close to zero.

To address this issue, motivated by the robustness of HOG (histogram of gradient features) in object recognition [Dalal and Triggs, 2005], we propose *linear pooling variables* (or pooling variables for short). They are basically regular HOG but save a normalization stage, so that linearity is preserved. By linearity, we mean that each pooling variable can be viewed as a linear filter applied to our edge labels. By pooling information from a neighborhood, these variables are more robust to local deformation. Due to their linearity, the value of our pooling variables are linear with respect to the labels of edge hypotheses (c.f. Eq (5.15)). Benefits of having linear pooling variables during our inference stage will be explained in Section 6.3.

5.2.3 Top-level latent variables

At top level, GPLVM is employed to obtain a low-dimensional latent representation of edge maps. The underlying assumption of GPLVM is that input features (the pooling variables in our case) follow a Gaussian distribution, with its mean and variance determined by latent variables. Since latent variables can vary smoothly in the latent spaces, GPLVM provides a continuum of templates of edge maps for objects from the same class. To give the reader a rough idea of what latent space may look like, Figure 5.4 illustrates some GPLVM-produced low-dimensional latent representations.

5.3 Conditional Random Field formulation

This section gives the detailed design of our framework, basing on a three-layered CRF model; each layer captures a particular level of the shape representation, as discussed in Section 5.2.

Let the binary vector $\mathbf{y} = \{y_i \mid i = 1...N_y\}$ denote all labels of the N_y edge hypotheses, and vector $\mathbf{t}$ denote all bottom-up observations including edge saliency and segment lengthes. M pooling variables are represented as a single M-dimensional vector $\mathbf{h}$. The low-dimensional latent variable is denoted as $\mathbf{x}$. Then, the CRF distribution is determined by its energy function:

$$P(\mathbf{y}, \mathbf{h}, \mathbf{x} \mid \mathbf{t}) \propto \exp\left(-E(\mathbf{y}, \mathbf{h}, \mathbf{x}, \mathbf{t})\right). \tag{5.1}$$

The energy function E consists of a bottom-level term E_B, a mid-level term E_M and a prior term E_P.

$$E(\mathbf{y}, \mathbf{h}, \mathbf{x}, \mathbf{t}) = E_B(\mathbf{y}, \mathbf{t}) + E_M(\mathbf{y}, \mathbf{h}, \mathbf{t}) + E_P(\mathbf{h}, \mathbf{x}) \tag{5.2}$$

5.3.1 The bottom-level term E_B

The bottom-level term is defined as follows:

$$E_B(\mathbf{y}, \mathbf{t}) = \sum_{i=1}^{N_y} v_i y_i + \sum_{(i,j)\in A} g_{ij} y_i y_j \tag{5.3}$$

The linear terms favor edges with strong local contrast. Parameter v_i denotes the cost of edge i, computed as $v_i = (1 - Pb_i)l_i$, where Pb_i is the average edge probability [Arbelaez et al., 2011], and l_i is the edge length. The quadratic terms enforce local inhibition between two adjacent and parallel edge hypotheses in the weakly supervised setting. The set A denotes pairs of adjacent edges. Parameter g_{ij} encodes the strength of inhibition between edge i and edge j. It is computed to be proportional to the completion energy of two edges [**?**]. The quadratic terms are not used in the supervised setting because bottom-up edge maps have been sparsified [Arbelaez et al., 2011].

5.3.2 The mid-level term E_M

To improve robustness to local deformations, a pooling variable is constrained to be the weighted sum of neighbouring edge labels:

$$\mathbf{h} = W\mathbf{y} \tag{5.4}$$

The matrix entry $W(i, j)$ is the weight of the vote from edge j to the pooling variable i. It is computed as the sum of votes from each edge point p in edge segment j to the variable:

$W(i,j) = \sum_{p \in j} \exp(-\beta_1 d_{ip}^2 - \beta_2 \phi_{ip}^2)$, where β_1 and β_2 are parameters. d_{ip} and ϕ_{ip} denote the distance and the angular difference between bin i and edge point p, respectively. The idea is that votes from neighbouring edges with similar orientations have larger weights. The purpose of the mid-level term is to ensure that the following linear relation strictly holds:

$$E_M(\mathbf{h}, \mathbf{y}, \mathbf{t}) = \delta(||\mathbf{h} - W\mathbf{y}||_2), \tag{5.5}$$

where $\delta(x) = 0$ if $x = 0$, otherwise $\delta(x) = \inf$.

5.3.3 The top-level prior term E_P

Built upon the mid-level representation, GPLVM is employed to capture intra-category shape variations. Let H denote the N by M feature matrix formed by stacking N features (pooling variables) of training samples, and let X denote the N by q latent variables matrix. Every row of X is a q dimensional vector corresponding to one training sample. The prior term E_P is chosen to be proportional to the negative logarithm of $P(\mathbf{h}, \mathbf{x} \mid H, X, \Theta)$, defined as:

$$P(\mathbf{h}, \mathbf{x} \mid H, X, \Theta) = P(\mathbf{h} \mid \mathbf{x}, H, X, \Theta)\, P(\mathbf{x} \mid H, X, \Theta), \tag{5.6}$$

where Θ denotes all the parameters.

The prior on $\mathbf{x}$ is simply an isotropic Gaussian distribution [Lawrence, 2005]:

$$P(\mathbf{x} \mid H, X, \Theta) = N(\mathbf{x} \mid 0, I_q) \tag{5.7}$$

The conditional distribution of $\mathbf{h}$ is also a Gaussian distribution,

$$P(\mathbf{h} \mid \mathbf{x}, H, X, \Theta) = N(\mathbf{h} \mid \mathbf{u}(\mathbf{x}), \sigma(\mathbf{x})^2 I_N), \tag{5.8}$$

where the parameter Θ in $\mathbf{u}$ and σ is omitted for clarity.

According to [Lawrence, 2005], the mean $\mathbf{u}$ is determined by the latent variable $\mathbf{x}$ and the training samples:

$$\mathbf{u}(\mathbf{x}) = H^T K(X)^{-1} k(X, \mathbf{x}) \tag{5.9}$$

The N by N matrix $K(X)$ encodes all the affinities between data points computed by kernel function $k(\cdot, \cdot)$. The parameter Θ in $K(X)$ is omitted.

The vector $k(X, \mathbf{x})$ has N rows, the value of row i is $k(\mathbf{x}_i, \mathbf{x})$ where $\mathbf{x}_i$ denotes the i-th row

of X. Variance σ^2 is

$$\sigma(\mathbf{x})^2 = k(\mathbf{x},\mathbf{x}) - k(X,\mathbf{x})^T K(X)^{-1} k(X,\mathbf{x}). \tag{5.10}$$

Overall, the prior term is defined as follows:

$$E_P(\mathbf{h},\mathbf{x}) = \frac{1}{2}\sigma(\mathbf{x})^{-2}||\mathbf{h}-\mathbf{u}(\mathbf{x})||_2^2 + MN\ln\sigma(\mathbf{x}) + \frac{1}{2}||\mathbf{x}||_2^2 \tag{5.11}$$

The first term penalizes deviation from the mean; the second term penalizes large variance; the third term encourages small latent variable values.

5.4 Maximal likelihood learning for GPLVM

To train a GPLVM model for a new category, we need to prepare the pooling features. When hand-labeled edge maps are available, all the edge labels are set to one and pooling variables are computed by Eq 5.15 as input features. In the weakly supervised case, gPb edges are first extracted from all input images, and their edge labels are set to be their gPb values. Then, the pooling variables are computed again by Eq 5.15. The pooling variables computed from gPb have some noise. However, the prediction of Gaussian process can suppress noise by pooling information from all data points. After obtaining the feature matrix H, the objective of learning is to determine the latent variables X and kernel parameters.

Our method uses maximal likelihood learning. Let $\mathbf{h}^j$ denote the j-th pooling variables of all samples. GPLVM assumes that the distributions of different variables are conditionally independent given the latent variables.

$$P(H \mid X,\Theta) = \prod_{j=1}^{M} P(\mathbf{h}^j \mid X,\Theta) \tag{5.12}$$

GPLVM assumes that $\mathbf{h}^j$ is from a Gaussian distribution:

$$P(\mathbf{h}^j \mid X,\Theta) = N(\mathbf{h}^j \mid 0, K(X)) \tag{5.13}$$

The log-likelihood of H is

$$L = -\frac{MN}{2}\ln 2\pi - \frac{M}{2}\ln|K| - \frac{1}{2}\mathrm{tr}(K(X)^{-1}HH^T) \tag{5.14}$$

We choose the particular form of RBF kernel in [Lawrence, 2005]. To learn the latent variables and its parameters, we maximize the log-likelihood Eq (5.14) with respect to these variables.

The software package in [Lawrence, 2005] is used.

5.4.1 The mid-level term E_M

To improve robustness to local deformations and simplify the inference problem, a HOG variable is constrained to be the weighted sum of neighbouring edge labels:

$$\mathbf{h} = W\mathbf{y} \tag{5.15}$$

The matrix entry $W(i,j)$ is the weight of the vote from edge j to HOG i. Votes from neighbouring edges with similar orientations have larger weights. The purpose of the mid-level term is to ensure the linear relation strictly holds:

$$E_M(\mathbf{h},\mathbf{y},\mathbf{t}) = \delta(||\mathbf{h} - W\mathbf{y}||_2), \tag{5.16}$$

where δ denotes Dirac delta function.

5.4.2 The top-level prior term E_P

Built upon mid-level representation, GPLVM is employed to capture intra-category shape variations. Let H denote the N by M feature matrix formed by stacking N features of training samples, and let X denote the N by q latent variables matrix. Every row of the latent variable matrix is a q dimensional latent vector corresponding to one training sample. The prior term E_P is chosen to be proportional to the negative logarithm of probability of $P(\mathbf{h},\mathbf{x} \mid H,X,\Theta)$ defined as:

$$P(\mathbf{h},\mathbf{x} \mid H,X,\Theta) = P(\mathbf{h} \mid \mathbf{x},H,X,\Theta)\, P(\mathbf{x} \mid H,X,\Theta), \tag{5.17}$$

where Θ denotes all the parameters.

The prior on $\mathbf{x}$ is simply an isotropic Gaussian distribution [Lawrence, 2005]:

$$P(\mathbf{x} \mid H,X,\Theta) = N(\mathbf{x} \mid 0, I_q) \tag{5.18}$$

The conditional distribution of $\mathbf{h}$ is also a Gaussian distribution,

$$P(\mathbf{h} \mid \mathbf{x},H,X,\Theta) = N(\mathbf{h} \mid \mathbf{u}(\mathbf{x}),\sigma(\mathbf{x})^2 I_N), \tag{5.19}$$

where the parameter Θ in $\mathbf{u}$ and σ is omitted for clarity.

According to [Lawrence, 2005], the mean $\mathbf{u}$ is determined by the latent variable $\mathbf{x}$ and the

training samples:

$$\mathbf{u}(\mathbf{x}) = H^T K(X)^{-1} k(X, \mathbf{x}) \tag{5.20}$$

The N by N matrix $K(X)$ encodes all the affinities between data points computed by kernel function $k(\cdot, \cdot)$. The parameter Θ in $K(X)$ is omitted.

The vector $k(X, \mathbf{x})$ has N rows, the value of row i is $k(\mathbf{x}_i, \mathbf{x})$ where $\mathbf{x}_i$ denotes the i-th row of X. Variance σ^2 is

$$\sigma(\mathbf{x})^2 = k(\mathbf{x}, \mathbf{x}) - k(X, \mathbf{x})^T K(X)^{-1} k(X, \mathbf{x}). \tag{5.21}$$

Overall, the prior term is defined as follows:

$$E_P(\mathbf{h}, \mathbf{x}) = \frac{1}{2}\sigma(\mathbf{x})^{-2}||\mathbf{h} - \mathbf{u}(\mathbf{x})||_2^2 + MN \ln \sigma(\mathbf{x}) + \frac{1}{2}||\mathbf{x}||_2^2 \tag{5.22}$$

The first term penalizes deviation from the mean; the second term penalizes large variance; the third term encourages small latent variable value.

5.5 An efficient inference method

For a test image, a CRF is set up as discussed in Section 5.3 using a learned prior term detailed in Section 5.4. This section discusses how to find the optimal edge labels of this CRF.

5.5.1 The supervised case

For a test image, both the latent variable $\mathbf{x}$ and the edge labels $\mathbf{y}$ are unknown. Marginalization of $\mathbf{x}$ is intractable due to nonlinearity in Eq (5.20) and Eq (5.21). Therefore, our method optimizes both $\mathbf{x}$ and $\mathbf{y}$. Because the constraint $\mathbf{h} = W\mathbf{y}$ is enforced by E_M, $\mathbf{h}$ can be replaced by $W\mathbf{y}$. Then, substituting Eq (5.22) and Eq (5.3) into Eq (5.2) leads to the following problem:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & \sum_{i=1}^{N_y} v_i y_i + MN \ln \sigma(\mathbf{x}) + \frac{1}{2}||\mathbf{x}||_2^2 \\ & + \frac{1}{2}\sigma(\mathbf{x})^{-2}||W\mathbf{y} - u(\mathbf{x})||_2^2 \qquad (5.23) \\ s.t. \quad & y_i \in \{0, 1\}, \forall i \qquad (5.24) \end{aligned}$$

The problem (5.23) is a high dimensional non-convex optimization problem. Since an image may have thousands of edge hypotheses, it is not efficient to optimize all variables jointly. We propose an efficient method to find a local optimal solution. First, the integer constraints on $\mathbf{y}$

are relaxed into linear ones: $0 \leq y_i \leq 1, \forall i$. Then the optimization problem is decomposed into bottom-up and top-down steps. In a bottom-up step, $\mathbf{y}$ is fixed, and $\mathbf{x}$ is optimized. In a top-down step, $\mathbf{x}$ is fixed and $\mathbf{y}$ is optimized. The bottom-up step is non-convex but only involves one variable with low dimensionality. The top-down step is a quadratic program. Therefore, both steps can be carried out efficiently. In experiments, the inference usually converges in a few iterations. The solution is typically not binary and is used as soft edge labels. Note that the top-down step will not be a quadratic program if $\mathbf{y}$ and $\mathbf{h}$ are not related in a simple linear way.

5.5.2 The weakly supervised case

Although sketching images without groundtruth edge maps for training is more challenging, the positive side is that the workload of hand-labeling is reduced. One only needs to collect images with roughly aligned objects. Different from the supervised case, all input images are involved in training and testing phases. In the learning phase, latent variables of all images are obtained. Therefore, the energy function (5.2) only depends on $\mathbf{y}$ and $\mathbf{h}$. Because of E_M, $\mathbf{h}$ can be replaced by $W\mathbf{y}$. By substituting Eq (5.22) and Eq (5.3) into Eq (5.2), and eliminating constants, we obtain:

$$\min_{\mathbf{y}} \sum_{i=1}^{N_y} v_i y_i + \sum_{(i,j)\in A} g_{ij} y_i y_j + ||W\mathbf{y} - \mathbf{u}||_2^2 / 2\sigma^2 \tag{5.25}$$

$$s.t. \quad y_i \in \{0,1\}, \forall i \tag{5.26}$$

where $\mathbf{u}$ and σ are computed by Eq (5.20) and Eq (5.21), respectively. After linear relaxation, our energy minimization problem reduces to a quadratic program which can be solved efficiently.

5.6 Experiments: supervised case

This section tests the supervised scenario in which category-specific information is learned from groundtruth contours. Using the cup and teapot datasets from [Wu et al., 2010c], we visualize key concepts of our method such as the latent space. Besides, we demonstrate that top-down information leads to better results than those by a bottom-up edge detector. On the Weizmann horse dataset from [Borenstein and Ullman, 2002b], our method is compared with two methods which also used top-down information for contour detection. Detection accuracy is reported against F-measure (F-value index) [Arbelaez et al., 2011].

Figure 5.3: First row: input images. Second row: gPb results. Third row: mean pooling variables of inferred latent variables. Fourth row: our results.

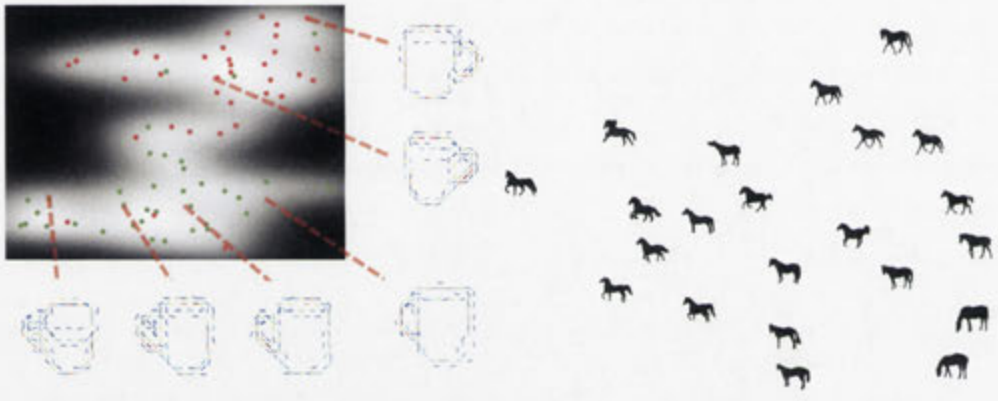


Figure 5.4: Left: The latent space of close-to-symmetry cups. Red dots denote the training data from images in the dataset. The green dots represent additional mirrored version of training data. The mean pooling features of sample points are displayed too. The obtained symmetry distribution of the latent points reflects the symmetry of our training data. Right: the latent space of horse contours. For clarity, shape masks are displayed instead of the contours.

In our implementation, pooling bins are placed at every 10 pixels along two image dimensions, and every 20 degrees in orientation. All images in one category are resized to the same size (e.g. 150 by 150 pixels for cup and teapot images) before computing the pooling features. The total number of pooling dimensions is about 3000. These parameters are selected by experimenting on a validation set. A standard GPLVM learning method is applied [Lawrence, 2005].

5.6.1 Validation on the cup and teapot dataset

Our method is first validated on the cup and teapot dataset. These datasets are challenging

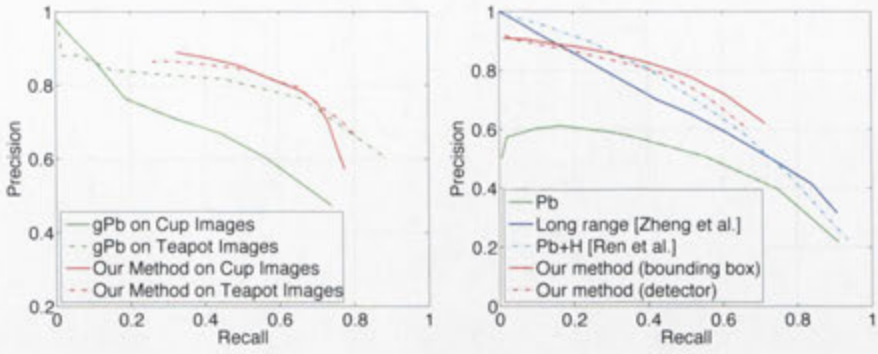


Figure 5.5: Left: The precision-recall curves of our method and gPb on the cup and teapot datasets, produced by the standard BSDS benchmarking algorithm [Arbelaez et al., 2011]. Right: The precision-recall curves of related methods on the Weizmann horse dataset. Our method is tested using either groundtruth bounding boxes or those from a detector [Ren and Ramanan, 2013]. The PR-curves of Pb, Zheng et al.'s method [Zheng et al., 2010] and Ren et al.'s method [Ren et al., 2005] are also shown when using high-level cues.

because there are a lot of distracting markings on cups or teapots while true contours (especially those of white cups) can be faint. Since our method uses gPb as the bottom-up edge detector on this dataset, we will use gPb as a baseline method to show that the top-down information can improve upon bottom-up edge detection results. These teapots and cups have different designs, sizes and poses. We have hand-labeled all groundtruth edge maps, and selected 36 cup images and 28 teapot images for training. To make a good use of limited data, a mirrored version of every training image is added. The dimensions of latent spaces of both datasets are set to two. Then we test the learned model on the rest 47 cup images and 32 teapot images.

Figure 5.3 shows that our results highlight object boundaries and suppress many of background contours and surface markings. On the cup dataset, our method achieves an F-value of 0.73 which is significantly higher than our input, gPb's result 0.58. On the teapot dataset, our method achieves 0.73 while gPb achieves 0.72. However, the PR curves show that our method has about 5% precision advantage over a large recall range (see Figure 5.5). On this dataset, our alternative inference algorithm converges in less than 10 iterations. Using pre-computed gPb results, it usually takes a couple of minutes to obtain one optimal edge map.

To show that our model captures global shape variations, the left of Figure 5.4 displays the learned latent space of the cup class. The red and green dots represent the original training data and their mirrored versions, respectively. The distribution of training data shows a symmetric pattern, suggesting that GPLVM correctly captures the relationship between training images. The mean pooling variables at some latent locations, computed by Eq (5.20) are also displayed. For clarity of visualization, each pooling variable is spasified by non-maximal suppression. We can see the mean pooling variables in the bottom gradually change from a cup to a mug.

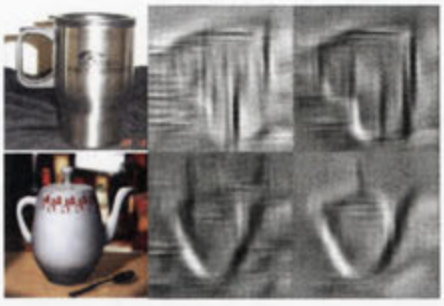


Figure 5.6: This figure shows that what kind of inverted images that we would see– if we look at a pooling map by "wearing" a Hoggles of [Vondrick et al., 2013]. From left to right: the original images; inverted images from the initial pooling variables; inverted images from the final pooling variables. Clearly, by suppressing category-irrelevant visual details, our method produces more meaningful inverted image at object categorical level.

Hoggles [Vondrick et al., 2013] may be used to produce interesting images by reverting pooling activations. Figure 5.6 compares the inverse images of pooling features derived from Pb and those from the optimal pooling variables. We can see that the latter are more recognizable, because the category-specific information has been preserved while irrelevant information has been suppressed.

5.6.2 Comparisons on the Weizmann horse dataset

Our method is compared with two related methods on the Weizmann horse dataset which has roughly-aligned horses of various poses in front of different background. Because our method does not solve the object detection problem, this dataset is more suitable than the datasets for which the object detection problem is quite challenging (e.g. ETH dataset used in [Ferrari et al., 2010]).

Training on the Weizmann horse dataset [Borenstein and Ullman, 2002b] is completed using 100 ground-truth edge maps. The dimension of latent space is set to 4 by experimenting on a validation set. The distribution of training data in the first two dimensions is shown in Figure 5.4. A gradual shape transition can be observed.

Because our method assumes that the objects are nearly aligned, we first crop image regions in the groundtruth bounding boxes, then compute optimal edge maps and project them back to the input images. On this dataset, our method uses Pb edge maps as input. Our results have an F-value of 0.67, while Pb achieves 0.54. Using the less accurate bounding boxes provided by a detector [Ren and Ramanan, 2013], our method still achieves an F-value of 0.63. Among the most related work [Ren et al., 2005] [Zheng et al., 2010] [Hariharan et al., 2011], we compare with [Ren et al., 2005] [Zheng et al., 2010] because their methods were also evaluated on this dataset, and precision-recall curves are shown in Figure 5.5.

This figure shows that our method has higher precision in mid-to-high recall range which is most useful for segmentation or recognition tasks. The work [Zheng et al., 2010] and [Ren et al., 2005] achieve F-values of 0.70 and 0.66, respectively. However, their methods use all local, mid-level and high-level cues while our method only uses local edge contrast and high-level shape cues. When not using mid-level cues, [Zheng et al., 2010] and [Ren et al., 2005] achieve F-values of 0.61 and 0.62, respectively. Considering (1) the local edge detector used in [Zheng et al., 2010] has 0.2 F-value advantage over Pb used by our method; (2) high level cues in [Ren et al., 2005] not only include shape cues but also include the texture familiarity cue which is effective for horses due to their distinctive colors, the comparison shows that our method is more accurate in modeling global shape variations.

5.7 Experiments: weakly supervised sketching

In [Wu et al., 2010c], the weakly supervised sketching problem is studied in the context of learning and recognition of object models. In this work, we focus on the visual quality of edge maps in terms of their likeness to the objects.

Our method sketches objects using a set of regularly spaced sticks comparable with those in ABM. Some results on two ABM datasets are shown in Figure 5.7. It can be seen that our method captures more faithful variations than ABM. ABM which learns a fixed template has difficulty with objects quite different from the mean shape. For example, their sketch for the last cup image in Figure 5.7 has incorrect orientation. We find 10 such errors for either the teapot dataset or the cup dataset while our method rarely makes one. All our results will be submitted as supplementary materials.

The F-measure of BSDS benchmark is based on points matching, which is sensitive to local deformation. However, the human vision system seems to be robust in this regard, considering that many hand-drawing sketches are not in perfect alignment with true object boundaries. To make F-measure robust to local deformation, we adapt it into a new metric called F-distance (FD for short) for comparing a sketch with a groundtruth edge map. It is based on the following definition. Recall-distance (RD) is defined as the average distance of a ground-truth edge point to its nearest edge point in the sketch. In a reverse way, precision-distance (PD) is defined as the average distance of a sketch point to its nearest edge point in the groundtruth map. FD is defined as the maximum of the mean PD value and the mean RD value on a dataset. Lower FD means higher quality. The FD values of our method and ABM on the cup dataset are 5.46 and 6.6 respectively. On the teapot dataset, FD values of these methods are 4.81 and 5.11, respectively. On both datasets, our method has better performance.

Figure 5.7: Comparison of sketches on the teapot dataset and the cup dataset.

5.8 Conclusion and future work

Detecting category specific edge maps is one of the basic functions of human perceptual grouping. To emulate this ability, many methods have been developed, from basic points matching to the complicated Bolzmann machine. Based on a non-parametric Bayesian method called GPLVM, we have developed a simple and principled CRF model. Our model is able to learn large shape variations while keep both learning and inference efficient. Superior edge maps are obtained as a result. Our findings suggest that the combination of conditional random field and the non-parametric Bayesian method is a promising approach to solve perceptual grouping problems. In future, we plan to extend our method to non-aligned natural scene images with the help of state-of-the-art object detectors. We are also interested in exploiting the latent variables for recognition.

Chapter 6

Symmetry Detection via Contour Grouping

6.1 Introduction

Bilaterally symmetric objects are abundant in the world, such as faces, leafs and architectures. Due to their importance to daily life, the human vision system has adapted well to detect symmetrical patterns. Perception of symmetry can influence many aspects of scene perception, such as figure-ground segmentation [Koffka, 1935][Reisfeld et al., 1995].

Our model focuses on estimation of symmetric axes of objects. Among all possible axes, our model aims to assign true symmetric axes with higher ranks. A key issue of symmetry detection is the diversity of objects. There are symmetric objects from different categories, having different shapes and texture. In order to accommodate various objects, our model needs to be flexible in two aspects. First, the model should use symmetry cues which are available in most of natural images. Second, the model should not be limited to any special kind of symmetric objects. For example, [Levinshtein et al., 2009a] is based on the statistics of symmetric parts, and is not suitable for the detection of large symmetric objects.

To meet these challenges, a symmetric object is modeled as a star-shaped graph connecting symmetric object parts. Each symmetric part is a pair of edgelets which are extracted from a contour image. Contour information is ubiquitous in natural images. If it can lead to an accurate estimation of the symmetric axes, our method can be applied to various images. In addition, all the symmetric pairs of edgelets are required to form a star graph. The star structure encodes the assumption that the symmetric axes of object parts will not deviate much from that of the whole object. Since this assumption is valid for general bilaterally symmetric objects (excluding those with curved symmetric axes), our model is not restricted to any special kind

of symmetric objects.

Our model is compared with Loy and Eklundh's method [Loy and Eklundh, 2006]. Despite being an early work, this method is still one of the best performing bottom-up grouping algorithms, according to the latest systematic evaluation [Park et al., 2008]. Our experiments show that our model achieves a better performance on two test datasets consisting of various natural scene images. The advantage comes from images containing abundant contour cues but scarce texture cues. These results justify the use of contour information. Our model can contribute to vision processes such as visual saliency [Kootstra et al., 2008] and symmetry-based segmentation [Sun and Bhanu, 2009; Cho and Lee, 2009; Gupta et al., 2005].

6.1.1 Problem setup and chapter overview

For an input image, our method estimates multiple symmetric axes, ranked according to their saliency. Our method is generic and does not depend on the category of the object. In Section 6.2, we introduce the graph of contextual interaction, which is constructed as a representation for symmetric objects. Section 6.3 discusses how our method extracts symmetric axes from the graph. Experimental evaluations are given in Section 6.4, and conclusion is drawn in Section 6.5.

6.2 A graph of contextual interaction

6.2.1 Method overview

A sample input image is shown in the top left of Figure 6.1, it has a salient symmetric axis in the middle. A number of isosceles trapezoids [Stahl and Wang, 2008] are extracted from pairs of edge segments as grouping elements. Then, a directed graph of contextual interaction is build. Each symmetric element (trapezoid) is represented by a node in the graph. Each node's preferred symmetric axis is the angle bisector of the legs of the trapezoid. A symmetric object usually consists of more than one element. Due to the principle of non-accidentalness, two elements will enhance each other if their symmetric axes are collinear. The contextual interaction of these elements is modeled by pairwise directed edges between nodes (the bottom left of Figure 6.1). Finally, the image's symmetric axis is detected by finding the most salient subgraph, shown in the bottom right figure.

6.2.2 Symmetric element extraction

In our model, symmetric elements are trapezoids made of line segment pairs. The local edge detector Pb [Arbelaez et al., 2011] is applied to an input image to obtain a soft contour image,

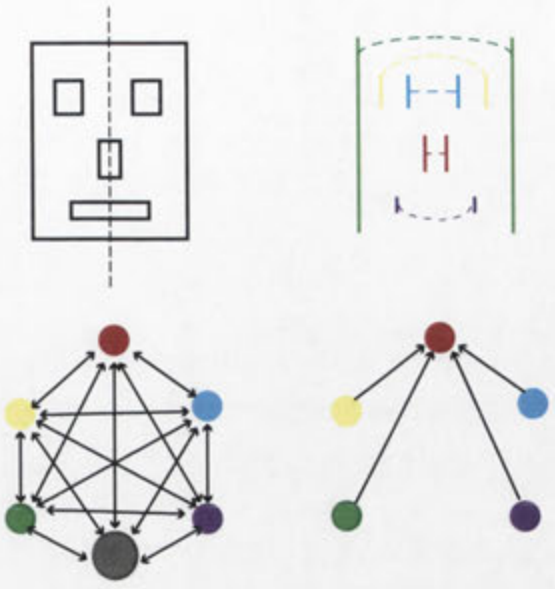


Figure 6.1: An illustration of our method. **Top left**: A line drawing image of a face. The dashed line is the salient symmetric axis human perceive. **Top right**: Five pairs of edgelets supporting the perceived symmetric axis. **Bottom left**: A graph of contextual interaction. Every colored node corresponds to the edge pair in same color in top right figure. The rest of edge pairs are denoted as a big gray node for clarity. The edges encode mutual enhancement of symmetric saliency. **Bottom right**: A star subgraph our model extracts.

in which the intensity of each pixel is its probability of being a true contour point. After thresholding, a line-fitting algorithm is used to extract line segments from the binary edge image [**?**]. The probability of each line segment is computed as the average Pb value of the associated points. Next, a trapezoid is extracted from two line segments by projecting line segments to the angle-bisector line and removing non-overlapping segments, as detailed in [Stahl and Wang, 2008]. The symmetric axis of a trapezoid is the angle-bisector line between these two line segments under the assumption of Euclidean transformation. The weight of a trapezoid w_i is designed as:

$$w_i = (Pb_{i1} Pb_{i2})^{\frac{1}{2}} \left(\frac{NCC_i + 1}{2} \right)^2 \tag{6.1}$$

where Pb_{i1} and Pb_{i2} denote the Pb values of two line segments respectively, and NCC_i denotes the normalized cross correlation of the left and flipped right half of gray scale images in the trapezoid. Using region information would reduce the weights of false matches. In sum, Eq 6.1 assigns higher weights to trapezoids formed by a symmetric region and salient contours.

6.2.3 Linking nodes with directed edges

Our model adds two directed edges between a pair of nodes if their symmetric axes are close enough. The weight of the directed edge e_{ij} reflects how much the trapezoid i could enhance the symmetric saliency induced by the trapezoid j. It is designed as follows:

$$w_{ij} = w_i g_{ij} \tag{6.2}$$

where g_{ij} reflects their geometric consistency and has a value between zero and one. This equation says the strength of enhancement is proportional to the saliency of node i, and is modulated by their geometric relationship, which is in turn defined as:

$$g_{ij} = \cos(\Delta\theta_{ij}) \exp\left(-\frac{d_{ij}}{\sigma_1}\right) \exp\left(-\frac{d_{ij}}{\sigma_2 m_i}\right) \tag{6.3}$$

where $\Delta\theta_{ij}$ is the angle between the symmetric axes of node i and node j. The d_{ij} denotes the distance from the center of trapezoid i to the symmetric axis j. The m_i is the length of the midline of trapezoid i. The parameters σ_1 and σ_2 control the amount of penalty. The first term $\cos(\Delta\theta_{ij})$ penalizes the angular difference of two axes. The second term penalizes the displacement of two symmetric axes. The third term penalizes the ratio of the displacement to the length of the midline of trapezoid i. These three terms capture the intuition that mutual enhancement is stronger when two trapezoids share the same symmetric axis.

6.3 Symmetric objects as star subgraphs

Our model assumes that each symmetric object consists of many symmetric elements i.e. trapezoids. We further assume that these elements are grouped together because they all add up to the saliency of a central element. Therefore, a salient symmetric object shall be represented as a start subgraph in the graph of contextual interaction. The star subgraph is chosen over subgraphs of other topologies, e.g. a chain or a tree subgraph for two reasons [Ishikawa et al., 2005][Liu et al., 2010b]. First, the topology of star subgraph ensures that all leaf nodes' preferred symmetric axes are close to that of the central node, thus ensuring the consistency of all object parts. However, the nodes at two ends of a long chain or tree graph may prefer very different axes. Second, finding a star subgraph leads to a much simpler optimization problem which can be solved in polynomial time. To find salient symmetric objects, we need to find the star subgraph with maximal weight. In our model, the weight of a subgraph is defined as the sum of the weight of the central node and those of incoming directed edges.

The inference problem is solved by enumerating all the star-subgraphs with different cen-

ters. For each node, our algorithm calculates the sum of all the weights of incoming edges larger than a threshold. The result is the maximal weight of any star graph centered on this node. In order to find the global maximum, it is necessary to enumerate all the nodes as the center of the star graph. Therefore the complexity of our method is $o(N^2)$, where N is the number of nodes. To save runtime, our model only considers nodes with weights larger than a threshold as the center.

6.3.1 Extracting multiple symmetric axes

To extract multiple symmetric axes from an input image, our model relies on the prior that the weight of any star subgraph representing an object should be a local maximum in a Hough space, which is the parametric space for all symmetry axes [Duda and Hart, 1972]. Therefore, each salient star subgraph is first projected to a point in the Hough space. Its coordinates in the Hough space are the parameters of its preferred symmetric axis. The weight of the point is just the weight of the subgraph. Based on this prior, some candidate points are extracted in this Hough space by finding the local maxima. This non-maximum suppression operation eliminates false-positive subgraphs which share nodes with true positives. Finally, candidate axes are sorted by their weights as our final output.

6.4 Experiments

6.4.1 Evaluation method and implementation details

A groundtruth symmetric axis in datasets is represented by two endpoints of a line segment. Each detected axis is represented as a line. A detected axis is considered as a true positive if the angle between itself and the groundtruth axis is less than 10 degree and the distance from either endpoint to the detected axis is less than 20 pixels. The recall and precision rates are computed when the number of output axes per image varies from one to twenty. The recall rate is defined as the ratio of true positives over the number of all groundtruth axes. The precision is defined as the percentage of true positives over the number of detection. Together, these two curves faithfully reflect a method's ability to rank hypotheses, and are free from the interference of threshold choice.

The Pb threshold is 0.05. In Eq 6.3, we set $\sigma_1 = 20$, $\sigma_2 = 0.125$. To find local maxima in the Hough space, we choose a window which spans $d/20$ pixels, and 40 degrees, where d stands for the length of the image's diagonal axis. To reduce noise, edges whose weights are less than 2 are removed.

Figure 6.2: Some results on synthetic images. The first column shows the test images, the second column shows the Pb detection. Rest of the columns show three most salient axes detected by our model. The first three axes are shown in red, yellow, green respectively. The matched edgelets with large weights are linked by thin blue lines. Best seen on screen.

6.4.2 Experiments on synthetic images

First of all, our model is tested on synthetic images. Synthetic images usually have clear contours and accurate symmetry correspondence. They are suitable for testing whether our model can find the symmetric axes for ideal inputs. Some sample images and the most salient three axes detected by our model are shown in Figure 6.2. This figure also displays the matched edgelets whose weights are larger than a threshold. We can see that the detection results are accurate.

6.4.3 Comparisons on the PSU dataset

The first dataset is a subset of the PSU dataset with the groundtruth[1]. There are 51 images with 74 symmetric axes in total. The recall and precision curves of both models on the PSU dataset are shown in the left part of Figure 6.3. These curves show that our model has a higher or compatible performance in a large area. Figure 6.4 shows first three detected symmetric axes by our model for some images. Figure 6.5 compares some detection results by both methods. For the image in the first row, our model's output is better than Loy and Eklundh's because there is hardly any symmetric texture in this image. For the image in the second row, both models detect the axes of two bottles. The difference is that our model relies on the bottle outlines, and their results are inferred mainly from the texture on the bottles. For the image on the third row, Loy and Eklundh's algorithm's output is better, due to abundance of texture and scarcity of contours. Since two models critically depend on the availability of texture and contour cues, the question that which model is on average better depends on the comparison of the two cues for symmetry detection task. Clearly, this question cannot be answered without much more extensive test. However, our experiments demonstrate that the contour can provide

[1] http://vision.cse.psu.edu/research/symComp12/index.shtml

very rich symmetry information which may not be available from texture.

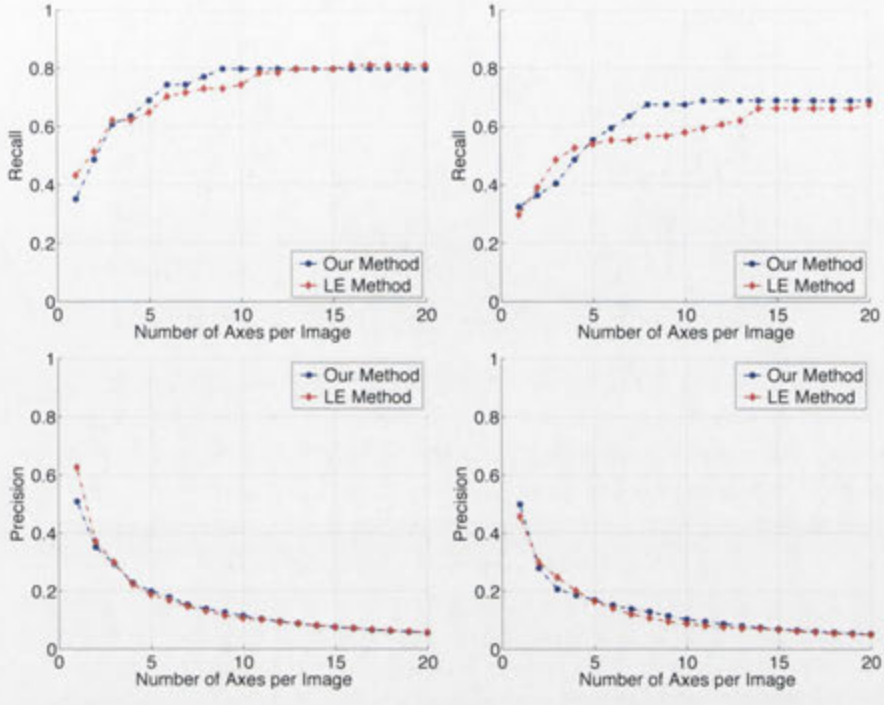


Figure 6.3: **Top left**: The recall rates of our model and Loy and Eklundh's method (LE for short) as a function of the number of output axes per image on the PSU dataset. **Top right**: the recall rates on the BSDS dataset. **Bottom left**: The precision curves on the PSU dataset. **Bottom right**: The precision curves on the BSDS dataset. Best seen on screen.

Figure 6.4: Our model's outputs for the PSU dataset images. Three most salient axes (in order of red, yellow, green) are shown.

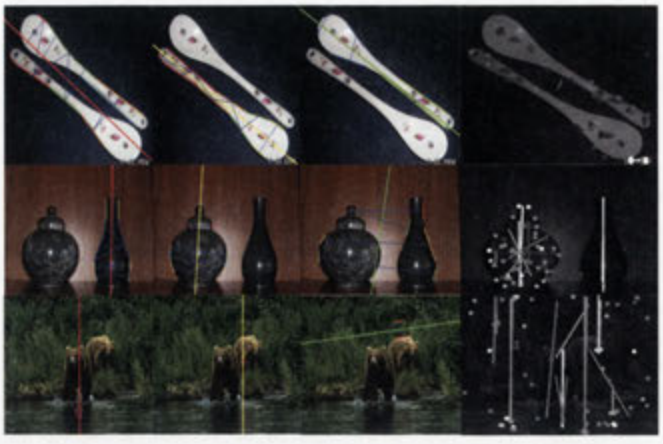


Figure 6.5: Comparisons with Loy and Eklundh's model [Loy and Eklundh, 2006] on the PSU dataset. The first three columns show the most salient axes our model detects and the last column shows the results by [Loy and Eklundh, 2006]. Our model does a better job on the spoon image in the first row, and [Loy and Eklundh, 2006] is better for the bear image in the last row. Both models correctly detect the axes in the second row image, but drawing on different cues. Best seen on screen.

6.4.4 Comparisons on the BSDS dataset

Next, we evaluate both models on the BSDS300 dataset, described by Arbelaez et al. [2011]. This image set is designed as a benchmark for image segmentation and contour detection. There are exemplary scenes in nature and cities in the dataset. Therefore, it is interesting to see how many of the typical scenes contain symmetric objects, and whether algorithms can perform well for images not having symmetry in mind when they are collected. We found around 50 images out of 200 images containing salient bilaterally symmetric objects. We hand-labeled the endpoints of symmetric axes. Some sample images and our labeling are shown in Figure 6.6. The some results are shown in Figure 6.7 where the five axes with highest saliency values are plotted. The recall and precision curves for both methods are shown in the right half of Figure 6.3. It shows that our model has compatible or slightly higher performance in most of the recall region. The performance of both models drops for the BSDS dataset, suggesting that these images are more challenging. Comparing with [Loy and Eklundh, 2006], the trend in the PSU dataset continues in the BSDS dataset. In Figure 6.8, our model produces better results for objects with long and clear outlines or markings.

6.5 Conclusion

This chapter proposes a bottom-up symmetry detection model based on grouping pairs of contour fragments. The contextual interaction of symmetric edgelets is represented as a directed

Figure 6.6: Some images containing symmetric objects selected from the BSDS300 dataset. The red line segments are the labeled symmetric axes.

Figure 6.7: Outputs of our model on the BSDS300 dataset.

Figure 6.8: Some images in the BSDS dataset for which our model produces better results. The first row is the results of LE method, and the second row shows our results.

graph. Then salient symmetric objects are extracted as star-shaped subgraphs with maximal weights. Compared with the SIFT-based method [Loy and Eklundh, 2006], our model is advantageous for images with scarce texture cues but clear contours. Our model can be used for applications such as symmetry-based segmentation and saliency detection.

Chapter 7

Conclusions and Future Directions

7.1 Summary

This thesis tackles the problem of contour grouping in natural scene images. Specifically, we investigate how to group contour fragments obtained from local edge detectors into complete object boundaries. Contour grouping is considered as a core part of the perceptual grouping process by psychologists. Their most inspiring studies of perceptual grouping are summarized by Gestalt principles. These principles capture human's preference to see certain visual patterns. However these principles are difficult to translate into computer algorithms. This thesis establishes suitable representations and algorithms to implement these principles based on machine learning techniques such as conditional random fields and the Gaussian process latent variable model. The contents of each chapters are summarized below:

Chapter 1 focuses on connecting contour fragments based on the Gestalt principle of good continuity and closure. Our main finding is that the closure principle can be effectively approximated by connectedness conditions. These connectedness conditions can be encoded by the closure potential functions of our conditional random field. The Gestalt principle of good continuity can also be encoded as cubic terms in our CRF. To solve the high order inference problem, we firstly reduce it to an integer program. Then a novel algorithm is devised to obtain a good solution of the integer program. Our experiments show that our method successfully connects contour fragments into long contours.

Chapter 2 addresses the detection of salient closed contours with the region cues. Region cues are wildly used in segmentation methods. In order to incorporate them into contour grouping methods, we argue that it is essential to guarantee the consistency between contour and region variables. Direct enforcing the consistency condition would involve non-linear constraints. However, we have found that based on the concept of winding numbers, some linear constraints are enough to ensure the consistency. Our experiments show that by incorporating

the region homogeneity cue, our method obtains improved results.

Chapter 3 presents a model for category specific contour grouping, drawing on the observation that past experience affects perceptual grouping. Our key contribution is a hierarchical model which represents both low-level edge maps and high level information, and has a mechanism for the top-down influence. At bottom level, edge maps are used to represent bottom-up edge detection. Mid-level information is represented as mid-level shape features which are more robust to local deformations. At top level, an edge map is represented by a latent variable with small dimensionality in a latent space. The familiarity of objects are coded in this latent space.

Chapter 4 discusses a symmetry detection algorithm which is solely based on contour information. The texture information is commonly used for symmetry detection, partly due to discriminative features such as SIFT. The contours on the other hand, are much less discriminative. Nevertheless, our method obtained comparable performance as a SIFT-based method, demonstrating the importance of contour information for this task.

7.2 Future Directions

In this section, we discuss some open problems and future directions.

7.2.1 Application of contour grouping

Contours are informative about the objects in images. Therefore it seems natural to use high quality contours for other computer vision problems. In Chapter 6, we show that contour cues can be as effective as texture cues for symmetry detection. However, we believe that high quality contours can be useful for high-level vision tasks such as recognition and object detection. Currently, the main stream methods for image classification are based on descriptors such as SIFT and HOG. These descriptors are basically histograms of image gradients which are quite noisy. Therefore, we conjuncture that using clean contours rather than noisy image gradients would improve accuracy. Contours also prove useful for object segmentation. In one of my on-going projects not discussed in this thesis, high-level classification information is integrated with mid-level contour information for object detection. This new method avoids the computational overhead of the sliding window approach, and improves on the state-of-the-art in terms of segmentation accuracy.

7.2.2 Temporal cues for contour grouping

Our thesis does not consider temporal cues for the contour grouping problem. However, if we want to extend our methods to videos, temporal cues cannot be overlooked. For example, the Gestalt principle of common motion states that elements which share the same motion pattern are more likely to be perceived as part of the same object. Also, some early psychological studies show that the temporal cues are developed at the earliest stages for infants, suggesting its importance in perceptual grouping. There are existing work about motion segmentation in videos. However, detecting general contours in videos, is a less understood problem. There are at least three aspects of temporal cues. The first one is the Gestalt principle of common motion. It summarizes the phenomenon that the perception of contours can be aroused by elements with different velocities. The second one is based on the temporal continuity of contours. Contours are part of physical objects which are stable in general. Assuming that videos have sufficient high sampling rates, contours should not change drastically between frames. The third one is based on the long term correlation of object contours. Certain object shapes may appear more than one time through out a video, e.g. a video of running horses. The contours of a horse in one frame may help detect the horse contours in other frames in which the horse is in a similar pose. We believe that the integration of all these temporal cues and other Gestalt cues could lead to better methods for this problem.

7.2.3 A unified model for perceptual grouping

This thesis builds separate models for different contour grouping scenarios. However, it is more desirable to build a unifying model which represents all the Gestalt cues effectively. Given that it seems difficult enough to consider principles independently, is it possible to build a computational model to accommodate all of them? We believe that advance of machine learning techniques such as deep learning methods offers an opportunity for constructing a unified model. We learned in Chapter 5, that a hierarchical structure is beneficial to model category-specific high-level grouping information. This philosophy is the cornerstone of deep learning methods. Therefore, adapting deep networks to solve perceptual grouping problems can be fruitful.

7.2.4 Computer vision, what is next?

Computer vision is a relatively young field. Legend has it that Mavin Minsky once assigned it to a undergraduate student as a three-month summer project in 1966. From that time, researchers begin to recognize the complexities of computer vision problems. Early methods

tackled them by heuristics or by studying some simplified scenarios. These approaches turned out to be limited. Marr revolutionized this area with his celebrated paradigm of vision research. Marr believed that the key to solving computer vision problems was to find the right representations and constraints which were rooted in the physical properties of our world. His paradigm effectively compare computer vision to physics. For many vision problems such as face detection, however, it is difficult, if at all possible, to derive explicitly-formulated constraints. Later approaches resorted to machine learning techniques such as the support vector machine to learn soft constraints from data. The features/representations of these methods were still largely hand-crafted. The newest trend is to learn representations directly from data through a deep network. Traditionally, these networks were trained with supervised data. However, it turned out that using unsupervised data would position deep learning methods among the state-of-the-art. Deep networks in theory can be at least as good as human since the human neural system can be regarded as one instance of deep networks. However, this approach gives rise to many new scientific questions. One is that the intelligence gained by machine does not directly translate into human knowledge. Even if machine solves one problem perfectly, we may still do not know why. Besides, there is still little theory regarding the optimal architecture and learning method of deep networks.

7.3 Conclusions

This thesis addresses the problem of contour grouping in natural scene images. In previous chapters, we have investigated how to represent well known, yet mysterious Gestalt principles such as continuity, closure, region homogeneity and object familiarity in different computational models. We also present efficient learning and inference methods for these novel models. This thesis has demonstrated the effectiveness of our new models, through comparisons with other state-of-the-art methods.

Our quest for understanding, on the level of computational theories, of perceptual grouping proves to be a longstanding one. In this final chapter, some open problems are discussed and some future directions are given. We hope this thesis would provide inspirations for future research.

Proofs for two theorems in Chapter 3

A.1 Two proofs to the non-submodularity of our energy function

We gave two different versions of proof to the non-submodular of our energy function.

The first proof (i.e. Proposition 3) requires a stronger condition, which assumes the existence of two intersecting single-connected contours, while the second one (i.e. Proposition 4) requires a weaker condition. Both cases are valid for most of the natural images.

Proposition 3. *If there are at least two single-connected contours intersecting with each other, then the energy function in* (A.2) *is non-submodular.*

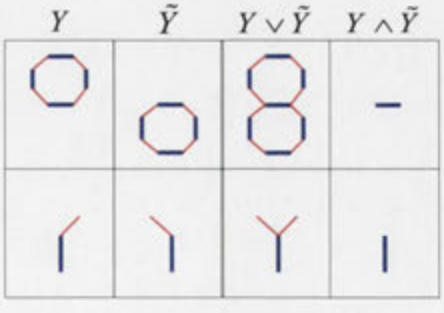


Figure A.1: Configurations used in the proofs. **First row**: Contours used in the proof for Proposition 1; **Second row**: Edgelets used in the proof of Proposition 2.

Proof. According to Eq(1) in our paper, the energy of our model is defined as:

$$E(\mathbf{Y},\mathbf{X}) = \sum_{i\in V_g} \phi_D\left(y_i,\mathbf{x}_i^d\right) + \sum_{q\in C^I} \psi_I\left(\mathbf{Y}_q,\mathbf{X}_q\right) \tag{A.1}$$

$$+ \sum_{q\in C^P\cup C^H} \psi_T\left(\mathbf{Y}_q\right) + \sum_{i\in V_c} \psi_M\left(y_i,\mathbf{x}_i^m\right) \tag{A.2}$$

Suppose $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ are two non-identical single connected contours which shares a few consecutive edges.[1] An example of these two contours and their union and intersection are shown in the first row of Figure A.1. Since $\mathbf{Y}$ and $\tilde{\mathbf{Y}}'$ are both connected contours which do not violate any constraints, there energy should be lower than any invalid configuration:

$$E(\mathbf{Y}) < M \tag{A.3}$$

$$E(\tilde{\mathbf{Y}}) < M, \tag{A.4}$$

where M is a very large positive cost for each violation of constraints. The pointwise maximum $\mathbf{Y}\vee\tilde{\mathbf{Y}}$ represents union of the two contours, which is again a valid connected contour. Therefore, we have

$$E\left(\mathbf{Y}\vee\tilde{\mathbf{Y}}\right) < M. \tag{A.5}$$

However, the point-wise minimum $\mathbf{Y}\wedge\tilde{\mathbf{Y}}$ is the shared part of two contours with two lose ends, i.e, two endpoints which do not connect to any completion edgelets. At each lose end, the extension constraint, i.e. Eq(5) in our paper is violated. As a result:

$$E\left(\mathbf{Y}\wedge\tilde{\mathbf{Y}}\right) > 2M. \tag{A.6}$$

Consequently,

$$E\Big(\mathbf{Y}\Big) + E\left(\tilde{\mathbf{Y}}\right) < E\left(\mathbf{Y}\vee\tilde{\mathbf{Y}}\right) + E\left(\mathbf{Y}\wedge\tilde{\mathbf{Y}}\right). \tag{A.7}$$

□

Now we move on to proof-version-2.

Proposition 4. *If there is one endpoint in boundary segment graph connected to at least two*

[1] From natural scene images, our model can usually extract hundreds of gradient edgelets, and completion edgelets are proposed abundantly, therefore, such two contours can be easily found.

completion edgelets and one gradient edgelet, the energy function in (A.2) *is non-submodular.*

Proof. Let $\mathbf{Y} = (1,1,0,0...0)^T$ and $\tilde{\mathbf{Y}} = (1,0,1,0...0)^T$. The first bit is the label of the gradient edgelet to which the endpoint is attached. The second and third bits are the labels of the two completion edgelets respectively. $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$ represents the configuration that the gradient edgelet is connected to either of the completion edgelet. All other edgelets are turned off. An example of these two contours and their union and intersection are shown in the second row of Figure A.1. Since the cost of violating constraints are much larger than the rest of the energy term, we have

$$E(\mathbf{Y}) \approx \sum_{q \in C^P \cup C^H} \psi_\Gamma(\mathbf{Y}_q) = M + M = 2M. \tag{A.8}$$

The first M comes from penalizing the lose end of the gradient edge, the second comes from penalizing one lose end of the completion edge. Similarly,

$$E(\tilde{\mathbf{Y}}) \approx 2M \tag{A.9}$$

The union $\mathbf{Y} \vee \tilde{\mathbf{Y}} = (1,1,1,0...0)^T$ will be penalized by three lose ends, each from one edgelets:

$$E(\mathbf{Y} \vee \tilde{\mathbf{Y}}) \approx 3M \tag{A.10}$$

The intersection $\mathbf{Y} \wedge \tilde{\mathbf{Y}} = (1,0,0,0...0)^T$ will be penalized by two lose ends of the gradient edgelet:

$$E(\mathbf{Y} \wedge \tilde{\mathbf{Y}}) \approx 2M. \tag{A.11}$$

Therefore,

$$E(\mathbf{Y} \vee \tilde{\mathbf{Y}}) + E(\mathbf{Y} \wedge \tilde{\mathbf{Y}}) - E\Big(\mathbf{Y}\Big) - E(\tilde{\mathbf{Y}}) \approx M. \tag{A.12}$$

□

A.2 Proof of feasibility

Proof. First, if the LPR and each MILP the algorithm needs to solve is feasible, our inference

can find a solution in at most $\left\lceil \frac{N}{N_{max}} \right\rceil$ iterations. where $\lceil x \rceil$ is the ceil function which return the minimum integer number larger or equal to x. N is the number of variables. N_{max} denotes the maximal number of integer dimensions that the subroutine can efficiently solve at one time. The solution, once obtained, satisfies the constraints (3.10)(3.11), thus is a feasible solution of the ILP. However, it is not obvious that the the LPR and MILP are feasible. The LPR is feasible because of the trivial solution $\mathbf{Y}^* = 0$. The feasibility of all MILP can be established by induction. Suppose Y^* is the solution of the previous MILP (or the LPR for the first iteration), We will show that the next MILP has at least one solution $\bar{\mathbf{Y}}^* = \lceil \mathbf{Y}^* \rceil$. In other words, $\bar{\mathbf{Y}}^*$ is obtained by any fractional label in Y^* to 1.

The feasibility of $\bar{\mathbf{Y}}^*$ can be established by checking the constraints of MILP. First of all, $\bar{\mathbf{Y}}^*$ is binary, therefore the Eq (3.17) is satisfied. Second, obviously $\bar{y}_i^* = y_i^*$ if $y_i^* \in \{0,1\}$. Therefore, the Eq (3.18) is satisfied. Finally, we show that $\bar{\mathbf{Y}}^*$ satisfies the inequalities. Since $\mathbf{Y}^*$ is the previous solution, these inequities are all satisfied:

$$y_i^* \leq y_j^*, \quad \forall i \in V_c, j \in V_g, (i,j) \in C^P. \tag{A.13}$$

$$y_j^* \leq \sum_{i \in q \cap V_c} y_i^*, \quad \forall j \in V_g, q \in C^\Gamma \tag{A.14}$$

$$0 \leq y_i^* \leq 1, \quad \forall i \in V \tag{A.15}$$

Eq(A.16)(A.18) are the connectedness constraints dissembled from Eq(3.19). It is straightforward to verify that

$$\lceil y_i^* \rceil \leq \lceil y_j^* \rceil, \quad \forall i \in V_c, j \in V_g, (i,j) \in C^P \tag{A.16}$$

$$\lceil y_j^* \rceil \leq \sum_{i \in q \cap V_c} \lceil y_i^* \rceil, \quad \forall j \in V_g, q \in C^\Gamma \tag{A.17}$$

$$0 \leq \lceil y_i^* \rceil \leq 1, \quad \forall i \in V \tag{A.18}$$

Therefore, $\bar{\mathbf{Y}}^*$ satisfies the Eq(3.16)(3.19) too. Thus, we have proved that MILP will always be feasible. Therefore, Algorithm 1 can always produce a solution. □

Bibliography

ALPERT, S.; GALUN, M.; BASRI, R.; AND BRANDT, A., 2007. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *Proc. CVPR*. (cited on pages 65 and 67)

ALPERT, S.; GALUN, M.; BRANDT, A.; AND BASRI, R., 2012. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE TPAMI*, 34, 2 (2012), 315–327. (cited on page 12)

ANDRES, B.; KAPPES, J. H.; BEIER, T.; KÖTHE, U.; AND HAMPRECHT, F., 2011. Probabilistic image segmentation with closedness constraints. In *Proc. International Conference on Computer Vision*. (cited on pages 11, 26, 51, and 52)

ANDREWS, S.; HAMARNEH, G.; AND SAAD, A., 2010. Fast random walker with priors using precomputation for interactive medical image segmentation. In *Lecture Notes in Computer Science, Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, III:9–16. (cited on page 71)

ARBELAEZ, P.; MAIRE, M.; FOWLKES, C.; AND MALIK, J., 2011. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33, 5 (2011), 898–916. (cited on pages xvi, xix, 2, 10, 12, 44, 63, 66, 78, 79, 80, 85, 87, 92, and 98)

BISHOP, C. M., 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. (cited on pages 11 and 15)

BORENSTEIN, E. AND ULLMAN, S., 2002a. Class-specific, top-down segmentation. In *Proc. European Conference on Computer Vision*, 109–124. (cited on pages xvi, 39, 44, and 65)

BORENSTEIN, E. AND ULLMAN, S., 2002b. Class-specific, top-down segmentation. In *Proc. ECCV*, 109–124. (cited on pages 85 and 88)

BOURDEV, L. D. AND MALIK, J., 2009. Poselets: Body part detectors trained using 3d human pose annotations. In *Proc. ICCV*, 1365–1372. (cited on page 13)

BOYD, S. AND VANDENBERGHE, L., 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA. ISBN 0521833787. (cited on page 64)

BOYKOV, Y.; VEKSLER, O.; AND ZABIH, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23 (2001), 1222–1239. (cited on pages 11, 16, 36, and 51)

BRESSON, X.; VANDERGHEYNST, P.; AND THIRAN, J.-P., 2006. A variational model for object segmentation using boundary information and shape prior driven by the mumford-shah functional. *International Journal of Computer Vision*, 68, 2 (2006), 145–162. (cited on page 12)

CANNY, J., 1986. A computational approach to edge-detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8, 6 (1986), 679–698. (cited on page 9)

CARREIRA, J. AND SMINCHISESCU, C., 2012. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE TPAMI*, 34, 7 (2012), 1312–1328. (cited on page 67)

CHO, M. AND LEE, K. M., 2009. Bilateral symmetry detection via symmetry-growing. In *Proc. BMVC*, 4.1–4.11. (cited on page 92)

COUR, T.; BENEZIT, F.; AND SHI, J., 2005. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE. (cited on page 20)

CREMERS, D., 2006. Dynamical statistical shape priors for level set-based tracking. *IEEE TPAMI.*, (2006), 1262–1273. (cited on pages 13 and 77)

CREMERS, D.; KOHLBERGER, T.; AND SCHNÖRR, C., 2003. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36, 9 (2003), 1929–1943. (cited on page 12)

DALAL, N. AND TRIGGS, B., 2005. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 886–893. (cited on page 79)

DELONG, A.; OSOKIN, A.; ISACK, H. N.; AND BOYKOV, Y., 2012. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96, 1 (2012), 1–27. (cited on page 35)

DUDA, R. O. AND HART, P. E., 1972. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15 (1972), 11–15. (cited on page 95)

EITZ, M.; HILDEBRAND, K.; BOUBEKEUR, T.; AND ALEXA, M., 2011. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *Visualization and Computer Graphics, IEEE Transactions on*, 17, 11 (2011), 1624–1636. (cited on page 2)

EK, C. H.; TORR, P. H.; AND LAWRENCE, N. D., 2008. Gaussian process latent variable models for human pose estimation. In *Machine learning for multimodal interaction*, 132–143. Springer. (cited on page 13)

ELDER, J. H. AND ZUCKER, S. W., 1996. Computing contour closure. In *Proc. European Conference on Computer Vision*, 399–412. (cited on pages 10 and 11)

ESLAMI, S. M. A.; HEESS, N.; AND WINN, J. M., 2012. The shape boltzmann machine: A strong model of object shape. In *PROC. CVPR*, 406–413. (cited on pages 13 and 76)

FELZENSZWALB, P. AND MCALLESTER, D., 2006. A min-cover approach for finding salient curves. In *CVPRW*. (cited on page 44)

FELZENSZWALB, P. F. AND HUTTENLOCHER, D. P., 2004. Efficient graph-based image segmentation. *IJCV*, (2004), 167–181. (cited on page 11)

FERRARI, V.; JURIE, F.; AND SCHMID, C., 2010. From images to shape models for object detection. *International Journal of Computer Vision*, 87, 3 (2010), 284–303. (cited on pages 13 and 88)

FOWLKES, C.; BELONGIE, S.; CHUNG, F.; AND MALIK, J., 2004. Spectral grouping using the nystrom method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26, 2 (2004), 214–225. (cited on page 20)

GALLAGHER, A. C.; BATRA, D.; AND PARIKH, D., 2011. Inference for order reduction in markov random fields. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11. IEEE Computer Society, Washington, DC, USA. (cited on page 11)

GALLIER, J. AND XU, D., 2013. The fundamental group, orientability. In *A Guide to the Classification Theorem for Compact Surfaces*, vol. 9 of *Geometry and Computing*. Springer. (cited on pages 11 and 53)

GEMAN, D.; GEMAN, S.; GRAFFIGNE, C.; AND DONG, P., 1990. Boundary detection by constrained optimization. *IEEE Trans. Pattern Analyis and Machine Intelligence*, 12, 7 (1990), 609–628. (cited on page 26)

GEMAN, S. AND GEMAN, D., 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6, 6 (nov. 1984), 721 –741. (cited on page 26)

GOULD, S., 2012. Multiclass pixel labeling with non-local matching constraints. In *CVPR*. (cited on pages 6 and 10)

GRADY, L., 2006. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28, 11 (2006), 1768–1783. (cited on page 71)

GUO, C.-E.; ZHU, S.-C.; AND WU, Y. N., 2007. Primal sketch: Integrating structure and texture. *Comput. Vis. Image Underst.*, 106, 1 (2007), 5–19. (cited on page 9)

GUPTA, A.; PRASAD, V.; AND DAVIS, L., 2005. Extracting regions of symmetry. In *Proc. ICIP*, vol. 3, III – 133–6. (cited on page 92)

GUY, G. AND MEDIONI, G., 1993. Inferring global perceptual contours from local features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 786–787. (cited on page 10)

HAMMER, P.; HANSEN, P.; AND SIMEONE, B., 1984. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Mathematical Programming*, 28 (1984), 121–155. (cited on pages 16 and 36)

HARIHARAN, B.; ARBELAEZ, P.; BOURDEV, L.; MAJI, S.; AND MALIK, J., 2011. Semantic contours from inverse detectors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 991 –998. (cited on pages 12, 13, 76, and 88)

HARRIS, C. AND STEPHENS, M., 1988. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, 147–151. (cited on page 47)

HAUAGGE, D. C. AND SNAVELY, N., 2012. Image matching using local symmetry features. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 206–213. IEEE. (cited on page 13)

HE, X.; ZEMEL, R. S.; AND CARREIRA-PERPIÑÁN, M. Á., 2004. Multiscale conditional random fields for image labeling. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 695–702. (cited on page 11)

HINTON, G. E., 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14, 8 (2002), 1771–1800. (cited on page 17)

HORN, B. K. P., 1983. The curve of least energy. *ACM Trans. Math. Softw.*, 9, 4 (Dec. 1983), 441–460. (cited on page 10)

HUBEL, D. H. AND WIESEL, T. N., 1959. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148, 3 (1959), 574–591. (cited on page 4)

ISHIKAWA, H., 2009. Higher-order clique reduction in binary graph cut. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2993 –3000. (cited on page 11)

ISHIKAWA, H.; GEIGER, D.; AND COLE, R., 2005. Finding tree structures by grouping symmetries. In *Proc. ICCV*, 1132–1139. (cited on pages 14 and 94)

ISING, E., 1925. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31, 1 (1925), 253–258. (cited on page 15)

JACOBSON, A.; KAVAN, L.; ; AND SORKINE-HORNUNG, O., 2013. Robust inside-outside segmentation using generalized winding numbers. *ACM SIGGRAPH*, 32, 4 (2013), 33:1–33:12. (cited on page 12)

JAIN, A. K., 1989. *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. ISBN 0-13-336165-9. (cited on page 9)

KASS, M.; WITKIN, A.; AND TERZOPOULOS, D., 1987. Snakes - active contour models. *International Journel of Computer Vision*, 1, 4 (1987), 321–331. (cited on pages 10, 11, 51, and 56)

KENDALL, D. G., 1989. A survey of the statistical theory of shape. *Statistical Science*, (1989), 87–99. (cited on page 13)

KENNEDY, R.; GALLIER, J.; AND SHI, J. B., 2011. Contour cut: identifying salient contours in images by solving a hermitian eigenvalue problem. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2520–2527. (cited on page 44)

KOFFKA, K., 1935. *Principles of gestalt psychology*. Harcourt Brace. (cited on pages 1, 10, 13, and 91)

KOKKINOS, I., 2010a. Boundary detection using f-measure-, filter- and feature- (f3) boost. In *Proc. European Conference on Computer Vision*, 650–663. (cited on page 56)

KOKKINOS, I., 2010b. Highly accurate boundary detection and grouping. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2520–2527. (cited on pages 9, 11, 26, 44, and 64)

KOLLAR, D. AND FRIEDMAN, N., 2009. *Probabilistic graphical models: principles and techniques*. The MIT Press. (cited on page 14)

KOLMOGOROV, V., 2006. Convergent tree-reweighted message passing for energy minimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28, 10 (2006), 1568–1583. (cited on page 17)

KOOTSTRA, G.; NEDERVEEN, A.; AND BOER, B. D., 2008. Paying attention to symmetry. In *Proc. BMVC*, 1115–1125. (cited on page 92)

KOVACS, I. AND JULESZ, B., 11993l. A closed curve is much more than an incomplete one - effect of closure in figure ground segmentation. *PNAS,USA*, 90, 16 (11993l), 7495–7497. (cited on page 25)

KOVESI, P. Matlab and octave functions for computer vision and image processing. Http://www.csse.uwa.edu.au/ pk/research/matlabfns. (cited on page 29)

KOVESI, P., 1999. Image features from phase congruency. *Videre: Journal of Computer Vision Research, MIT Press*, (1999). (cited on page 9)

LADICKY, R.; RUSSELL, C.; KOHLI, P.; AND TORR, P., 2012. Inference methods for crfs with co-occurrence statistics. *International Journal of Computer Vision*, (2012), 1–13. (cited on page 10)

LAFFERTY, J.; MCCALLUM, A.; AND PEREIRA, F. C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001). (cited on pages 6 and 14)

LAWRENCE, N. D., 2005. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *JMLR*, 6 (2005), 1783–1816. (cited on pages 6, 13, 20, 76, 81, 82, 83, and 86)

LAWRENCE, N. D., 2007. Learning for larger datasets with the gaussian process latent variable model. In *AISTATS*, 243–250. (cited on page 21)

LEMPITSKY, V. S.; KOHLI, P.; ROTHER, C.; AND SHARP, T., 2009. Image segmentation with a bounding box prior. In *ICCV*, 277–284. (cited on page 11)

LEUNG, T. AND MALIK, J., 1998. Contour continuity in region based image segmentation. In *Proc. European Conference on Computer Vision*, 544 –559. (cited on pages 10, 12, and 52)

LEVENTON, M. E.; GRIMSON, W. E. L.; AND FAUGERAS, O. D., 2000. Statistical shape influence in geodesic active contours. In *Proc. CVPR*, 1316–1323. (cited on page 13)

LEVINSHTEIN, A.; DICKINSON, S.; AND SMINCHISESCU, C., 2009a. Multiscale symmetric part detection and grouping. In *Proc. ICCV*, 2162–2169. (cited on pages 14 and 91)

LEVINSHTEIN, A.; SMINCHISESCU, C.; AND DICKINSON, S., 2010a. Optimal contour closure by superpixel grouping. In *Proceedings of the 11th European conference on Computer vision*, 480–493. (cited on page 10)

LEVINSHTEIN, A.; SMINCHISESCU, C.; AND DICKINSON, S., 2010b. Optimal contour closure by superpixel grouping. In *Proc. ECCV*, 480–493. (cited on pages 12, 60, 62, 65, 66, and 67)

LEVINSHTEIN, A.; STERE, A.; KUTULAKOS, K.; FLEET, D.; DICKINSON, S.; AND SIDDIQI, K., 2009b. Turbopixels: Fast superpixels using geometric flows. *IEEE TPAMI*, 31, 12 (2009), 2290 –2297. (cited on page 56)

LI, Z., 1998. A neural model of contour integration in the primary visual cortex. *Neural Computation*, 10 (1998), 903–940. (cited on page 10)

LINDEBERG, T., 1994. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21, 2 (1994), 224–270. (cited on page 10)

LIU, C.; YUEN, P. C.; AND QIU, G., 2009. Object motion detection using information theoretic spatio-temporal saliency. *Pattern Recognition*, 42, 11 (2009), 2897–2906. (cited on page 2)

LIU, J. C. AND LIU, Y. X., 2010. Curved reflection symmetry detection with self-validation. In *Proc. ACCV*, 102–114. (cited on page 14)

LIU, Y.; HEL-OR, H.; KAPLAN, C. S.; AND GOOL, L. J. V., 2010a. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision*, 5, 1-2 (2010). (cited on page 13)

LIU, Y.; HEL-OR, H.; KAPLAN, C. S.; AND GOOL, L. J. V., 2010b. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision*, 5 (2010), 1–195. (cited on page 94)

LOY, G. AND EKLUNDH, J. O., 2006. Detecting symmetry and symmetric constellations of features. In *Proc. ECCV*, 508–521. (cited on pages xx, 14, 92, 98, and 100)

MAHAMUD, S.; WILLIAMS, L. R.; THORNBER, K. K.; AND XU, K. L., 2003. Segmentation of multiple salient closed contours from real images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25, 4 (2003), 433–444. (cited on pages 10 and 26)

MAIRE, M.; ARBELAEZ, P.; FOWLKES, C.; AND MALIK, J., 2008. Using contours to detect and localize junctions in natural images. *2008 IEEE Conference on CVPR, Vols 1-12*, (2008), 611–618. (cited on page 47)

MARR, D., 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc. (cited on pages 2 and 9)

MARR, D. AND HILDRETH, E., 1980. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207, 1167 (1980), 187–217. (cited on page 9)

MARTIN, D. R.; FOWLKES, C. C.; AND MALIK, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26, 5 (2004), 530–549. (cited on pages xv, 9, 10, 27, and 28)

MCINTYRE, M. AND CAIRNS, G., 1993. A new formula for winding number. *Geometriae Dedicata*, 46, 2 (1993), 149–159. (cited on page 11)

MEISTER, A. L. F., 1769. *Generalia de genesi figurarum planarum et inde pendentibus earum affectionibus*. (cited on page 52)

MING, Y.; LI, H.; AND HE, X., 2012. Connected contours: A new contour completion model that respects the closure effect. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 829 –836. (cited on pages 51 and 62)

MORI, G.; REN, X.; EFROS, A. A.; AND MALIK, J., 2004. Recovering human body configurations: combining segmentation and recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 326–333. (cited on pages xvi and 44)

NEEDHAM, T., 1999. *Visual Complex Analysis*. Oxford University Press, USA. (cited on pages 11 and 54)

NICOLLS, F. AND TORR, P. H., 2010. Discrete minimum ratio curves and surfaces. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2133–2140. IEEE. (cited on page 12)

NISS, M., 2009. History of the lenz-ising model 1950-1965: from irrelevance to relevance. *Archive for History of Exact Sciences*, 63 (2009), 243–287. (cited on page 10)

PALMER, S. E., 1999. *Vision science : photons to phenomenology*. MIT Press, Cambridge, Mass. (cited on pages 5, 10, 25, and 26)

PALMER, S. E.; BROOKS, J. L.; AND NELSON, R., 2003. When does grouping happen? *Acta Psychologica*, 114, 3 (2003), 311–330. (cited on pages xv, 3, 4, and 9)

PARENT, P. AND ZUCKER, S. W., 1989. Trace inference, curvature consistency, and curve detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11 (1989), 823–839. (cited on page 10)

PARK, M.; LEE, S.; CHEN, P. C.; KASHYAP, S.; BUTT, A. A.; AND LIU, Y. X., 2008. Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In *Proc. CVPR*, 3745–3752. (cited on page 92)

PEARL, J., 1982. Reverend bayes on inference engines: a distributed hierarchical approach. In *in Proceedings of the National Conference on Artificial Intelligence*, 133–136. (cited on page 17)

PIOTR DOLLAR, S. B., ZHUOWEN TU, 2006. Supervised learning of edges and object boundaries. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1964–1971. (cited on page 9)

PRASAD, V. S. N. AND YEGNANARAYANA, B., 2004. Finding axes of symmetry from potential fields. *IEEE TIP*, 13, 12 (2004), 1559–1566. (cited on page 14)

PRISACARIU, V. AND REID, I., 2011a. Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2185 –2192. (cited on page 12)

PRISACARIU, V. A. AND REID, I., 2011b. Nonlinear shape manifolds as shape priors in level set segmentation and tracking. In *Proc. CVPR*, 2185–2192. (cited on pages 13, 76, and 77)

RAND, W. M., 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66, 336 (1971), 846–850. (cited on page 44)

REISFELD, D.; WOLFSON, H.; AND YESHURUN, Y., 1995. Context-free attentional operators - the generalized symmetry transform. *IJCV*, 14, 2 (1995), 119–130. (cited on page 91)

REN, X., 2008. Multi-scale improves boundary detection in natural images. In *Proceedings of the 10th European Conference on Computer Vision*, 533–545. (cited on page 10)

REN, X. AND BO, L., 2012. Discriminatively trained sparse code gradients for contour detection. (cited on page 9)

REN, X.; MALIK, J.; AND FOWLKES, C. C., 2005. Cue integration for figure/ground labeling. In *NIPS*. (cited on pages xix, 12, 77, 87, 88, and 89)

REN, X. AND RAMANAN, D., 2013. Histograms of sparse codes for object detection. In *Proc. CVPR*. (cited on pages xix, 87, and 88)

REN, X. F.; FOWLKES, C. C.; AND MALIK, J., 2006. Figure/ground assignment in natural images. *Computer Vision - Eccv 2006, Pt 2, Proceedings*, 3952 (2006), 614–627. (cited on page 10)

REN, X. F.; FOWLKES, C. C.; AND MALIK, J., 2008. Learning probabilistic models for contour completion in natural images. *International Journel of Computer Vision*, 77, 1-3 (2008), 47–63. (cited on pages xvi, 10, 11, 26, 29, and 44)

ROTH, S. AND BLACK, M. J., 2005. Fields of experts: A framework for learning image priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 860–867. (cited on page 11)

ROTHER, C.; KOLMOGOROV, V.; AND BLAKE, A., 2004. "grabcut": interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, 309–314. (cited on pages 12 and 71)

ROTHER, C.; KOLMOGOROV, V.; LEMPITSKY, V.; AND SZUMMER, M., 2007. Optimizing binary mrfs via extended roof duality. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–8. IEEE. (cited on page 16)

SCHOENEMANN, T.; KAHL, F.; MASNOU, S.; AND CREMERS, D., 2012. A linear framework for region-based image segmentation and inpainting involving curvature penalization. *IJCV*, 99, 1 (2012), 53–68. (cited on page 12)

SCHOENEMANN, T.; MASNOU, S.; AND CREMERS, D., 2011. The elastic ratio: Introducing curvature into ratio-based image segmentation. *IEEE TIP*, (2011), 2565–2581. (cited on pages 10, 51, and 60)

SCHÖLKOPF, B., 2002. *Learning with kernels*. The MIT Press. (cited on page 20)

SCHÖLKOPF, B.; SMOLA, A.; AND MÜLLER, K.-R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10, 5 (Jul. 1998), 1299–1319. (cited on page 13)

SETHIAN, J. A., 1999. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, vol. 3. Cambridge university press. (cited on page 11)

SHA'ASUA, A. AND ULLMAN, S., 1988. Structural saliency: The detection of globally salient structures using a locally connected network. In *ICCV*, 321–327. (cited on page 10)

SHARON, E.; BRANDT, A.; AND BASRI, R., 2000. Completion energies and scale. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22, 10 (2000), 1117–1131. (cited on page 32)

SHI, J. AND MALIK, J., 2000a. Normalized cuts and image segmentation. *IEEE TPAMI*, 22 (2000), 888–905. (cited on pages 11, 51, 56, 65, and 66)

SHI, J. B. AND MALIK, J., 2000b. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22, 8 (2000), 888–905. (cited on pages 2, 18, 20, and 26)

SHOTTON, J.; BLAKE, A.; AND CIPOLLA, R., 2008. Multiscale categorical object recognition using contour fragments. *IEEE TPAMI*, 30, 7 (2008), 1270–1281. (cited on page 2)

STAHL, J. S. AND WANG, S., 2007. Edge grouping combining boundary and region information. *IEEE TIP*, 16, 10 (2007), 2590–2606. (cited on pages 12, 51, 52, 56, 60, 61, and 66)

STAHL, J. S. AND WANG, S., 2008. Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *IEEE TPAMI*, 30, 3 (2008), 395–411. (cited on pages 14, 51, 92, and 93)

SUEN LEE, M. AND MEDIONI, G., 1999. Grouping into regions, curves, and junctions. *Computer Vision Image Understanding*, (1999). (cited on page 10)

SUMENGEN, B. AND MANJUNATH, B. S. Graph partitioning active contours (gpac) for image segmentation. *IEEE TPAMI*, 509–521. (cited on pages 12, 51, and 66)

SUN, Y. AND BHANU, B., 2009. Symmetry integrated region-based image segmentation. In *Proc. CVPR*, 826–831. (cited on page 92)

SUN, Y. AND BHANU, B., 2012. Reflection symmetry-integrated image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34, 9 (2012), 1827–1841. (cited on page 13)

SUTTON, C. AND MCCALLUM, A., 2009. Piecewise training for structured prediction. *Machine Learning*, 77 (2009), 165–194. (cited on page 39)

TABB, M. AND AHUJA, N., 1997. Multiscale image segmentation by integrated edge and region detection. *IEEE TIP*, 6, 5 (may 1997), 642 –655. doi:10.1109/83.568922. (cited on page 12)

TASKAR, B.; CHATALBASHEV, V.; KOLLER, D.; AND GUESTRIN, C., 2005. Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd international conference on Machine learning*, 896–903. ACM. (cited on page 18)

TONG, W.-S. AND TANG, C.-K., 2005. Robust estimation of adaptive tensors of curvature by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27, 3 (2005), 434–449. (cited on page 10)

TSOGKAS, S. AND KOKKINOS, I., 2012. Learning-based symmetry detection in natural images. In *Computer Vision–ECCV 2012*, 41–54. Springer. (cited on page 13)

VICENTE, S.; KOLMOGOROV, V.; AND ROTHER, C., 2008. Graph cut based image segmentation with connectivity priors. In *Computer Vision and Pattern Recognition, 2008, IEEE Conference on*. (cited on page 11)

VINCENT, L. AND SOILLE, P., 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE transactions on pattern analysis and machine intelligence*, 13, 6 (1991), 583–598. (cited on page 11)

VONDRICK, C.; KHOSLA, A.; MALISIEWICZ, T.; AND TORRALBA, A., 2013. HOGgles: Visualizing Object Detection Features. *ICCV*, (2013). (cited on pages xix and 88)

WAINWRIGHT, M. J.; JAAKKOLA, T. S.; AND WILLSKY, A. S., 2005. Map estimation via agreement on trees: message-passing and linear programming. *Information Theory, IEEE Transactions on*, 51, 11 (2005), 3697–3717. (cited on page 17)

WALTHER, D. B.; CHAI, B.; CADDIGAN, E.; BECK, D. M.; AND FEI-FEI, L., 2011. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, 108, 23 (2011), 9661–9666. (cited on page 51)

WANG, S.; KUBOTA, T.; SISKIND, J. M.; AND WANG, J., 2005. Salient closed boundary extraction with ratio contour. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27, 4 (2005), 546–561. (cited on pages 10, 12, 26, 52, and 60)

WANG, S. AND SISKIND, J. M., 2003. Image segmentation with ratio cut. *IEEE TPAMI*, 25 (2003), 675–690. (cited on pages 11, 18, 19, 20, and 51)

WHITNEY, H., 1937. On regular closed curves in the plane. *Compositio Mathematica*, 4 (1937), 276–284. (cited on page 11)

WILLIAMS, L. R. AND THORNBER, K. K., 1999. A comparison of measures for detecting natural shapes in cluttered backgrounds. *International Journel of Computer Vision*, 34, 2-3 (1999), 81–96. (cited on page 10)

WOODFORD, O.; TORR, P.; REID, I.; AND FITZGIBBON, A., 2008. Global stereo reconstruction under second order smoothness priors. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1 –8. (cited on page 10)

WU, Y.; SI, Z.; GONG, H.; AND ZHU, S.-C., 2010a. Learning active basis model for object detection and recognition. *International Journal of Computer Vision*, 90 (2010), 198–235. (cited on page 12)

WU, Y. N.; SI, Z.; GONG, H.; AND ZHU, S.-C., 2010b. Learning active basis model for object detection and recognition. *Int. J. Comput. Vision*, 90, 2 (2010), 198–235. (cited on page 12)

WU, Y. N.; SI, Z.; GONG, H.; AND ZHU, S. C., 2010c. Learning active basis model for object detection and recognition. *IJCV*, 90, 2 (2010), 198–235. (cited on pages 13, 75, 76, 77, 85, and 89)

WU, Z. AND LEAHY, R., 1993. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15, 11 (1993), 1101–1113. (cited on pages 18 and 19)

XIE, X. AND MIRMEHDI, M., 2004. Rags: Region-aided geometric snake. *IEEE TIP*, (2004), 640–652. (cited on page 51)

YLAJAASKI, A. AND ADE, F., 1996. Grouping symmetrical structures for object segmentation and description. *Computer Vision and Image Understanding*, 63, 3 (1996), 399–417. (cited on page 14)

YU, S. X.; LEE, T. S.; AND KANADE, T., 2001. A hierarchical markov random field model for figure-ground segregation. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2134 (2001), 118–133. (cited on page 12)

ZHANG, L. AND JI, Q., 2010. Image segmentation with a unified graphical model. *IEEE TPAMI*, 32, 8 (2010), 1406–1425. (cited on page 52)

ZHENG, S.; YUILLE, A.; AND TU, Z., 2010. Detecting object boundaries using low-, mid-, and high-level information. *Computer Vision and Image Understanding*, 114, 10 (2010), 1055–1067. (cited on pages xix, 12, 77, 87, 88, and 89)

ZITNICK, C. AND PARIKH, D., 2012. The role of image understanding in contour detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 622 –629. (cited on pages 10 and 12)